A Survey of Reinforcement Learning for Large Reasoning Models

Kaiyan Zhang[1*†], Yuxin Zuo[1*†], Bingxiang He[1*], Youbang Sun[1*], Runze Liu[1*], Che Jiang[1*], Yuchen Fan[2,3*], Kai Tian[1*], Guoli Jia[1*], Pengfei Li[2,6*], Yu Fu[9*], Xingtai Lv[1*], Yuchen Zhang[2,4*], Sihang Zeng[7*], Shang Qu[1,2*], Haozhan Li[1*], Shijie Wang[2*], Yuru Wang[1*], Xinwei Long[1], Fangfu Liu[1], Xiang Xu[5], Jiaze Ma[1], Xuekai Zhu[3], Ermo Hua[1,2], Yihao Liu[1,2], Zonglin Li[2], Huayu Chen[1], Xiaoye Qu[2], Yafu Li[2], Weize Chen[1], Zhenzhao Yuan[1], Junqi Gao[6], Dong Li[6], Zhiyuan Ma[8], Ganqu Cui[2], Zhiyuan Liu[1], Biqing Qi[2‡], Ning Ding[1,2‡], Bowen Zhou[1,2‡]

[1] Tsinghua University   [2] Shanghai AI Laboratory   [3] Shanghai Jiao Tong University   [4] Peking University
[5] University of Science and Technology of China   [6] Harbin Institute of Technology   [7] University of Washington
[8] Huazhong University of Science and Technology   [9] University College London

[†] **Project Lead.**   [*] **Core Contributors.**   [‡] **Corresponding Authors.**

✉ zhang-ky22@mails.tsinghua.edu.cn   ⭘ TsinghuaC3I/Awesome-RL-for-LRMs

Abstract | In this paper, we survey recent advances in Reinforcement Learning (RL) for reasoning with Large Language Models (LLMs). RL has achieved remarkable success in advancing the frontier of LLM capabilities, particularly in addressing complex logical tasks such as mathematics and coding. As a result, RL has emerged as a foundational methodology for transforming LLMs into LRMs. With the rapid progress of the field, further scaling of RL for LRMs now faces foundational challenges not only in computational resources but also in algorithm design, training data, and infrastructure. To this end, it is timely to revisit the development of this domain, reassess its trajectory, and explore strategies to enhance the scalability of RL toward Artificial SuperIntelligence (ASI). In particular, we examine research applying RL to LLMs and LRMs for reasoning abilities, especially since the release of DeepSeek-R1, including foundational components, core problems, training resources, and downstream applications, to identify future opportunities and directions for this rapidly evolving area. We hope this review will promote future research on RL for broader reasoning models.
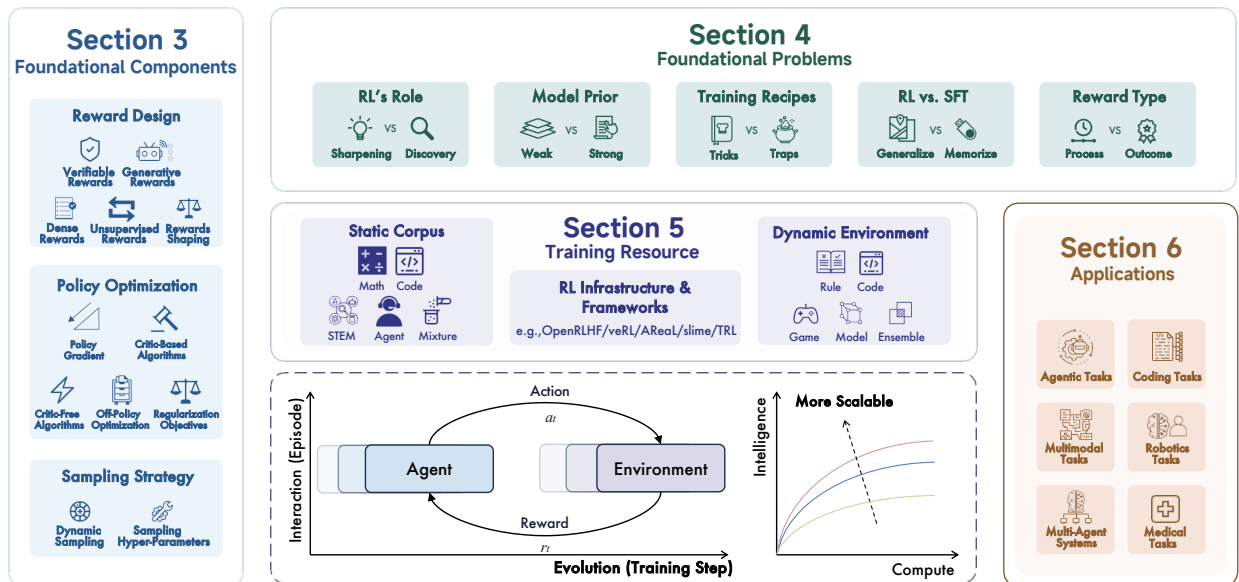
Figure 1 | Overview of the survey. We introduce the foundational components of RL for LRMs, along with open problems, training resources, and applications. Central to this survey is a focus on large-scale interactions between language agents and environments throughout long-term evolution.

Contents

1. Introduction

Reinforcement Learning (RL) [Sutton et al., 1998] has repeatedly demonstrated that narrow, well-specified reward signals can drive artificial agents to superhuman competence on complex tasks. Landmark systems such as AlphaGo [Silver et al., 2016] and AlphaZero [Silver et al., 2017], which learned exclusively through self-play and reward feedback, surpassed world champions in Go, chess, shogi and Stratego [Perolat et al., 2022, Schrittwieser et al., 2020, Silver et al., 2018], establishing RL as a practical and promising technology for high-level problem solving. In the era of Large Language Models (LLMs) [Zhao et al., 2023a], RL initially rose to prominence as a post-training strategy for human alignment [Ouyang et al., 2022]. Widely adopted methods such as Reinforcement Learning from Human Feedback (RLHF) [Christiano et al., 2017] and Direct Preference Optimization (DPO) [Rafailov et al., 2023] finetune pre-trained models to follow instructions and reflect human preferences, markedly improving helpfulness, honesty, and harmlessness (3H) [Bai et al., 2022b].

More recently, a new trend has emerged: RL for Large Reasoning Models (LRMs) [Xu et al., 2025a], which aims not merely to align behavior but to incentivize reasoning itself. Two recent milestones (i.e., OpenAI o1 [Jaech et al., 2024] and DeepSeek-R1 [Guo et al., 2025a]) demonstrate that training LLMs using reinforcement learning with verifiable rewards (RLVR), such as answer correctness for mathematics or unit-test pass rates for code, can enable models to perform long-form reasoning, including planning, reflection, and self-correction. OpenAI reports [Jaech et al., 2024] that o1's performance improves smoothly with both additional RL (increased train-time compute) and more time spent "thinking" at inference (test-time compute) [Brown et al., 2024, Liu et al., 2025l, Snell et al., 2024], revealing a new scaling axis beyond pre-training alone [Aghajanyan et al., 2023, Kaplan et al., 2020]. DeepSeek-R1 [Guo et al., 2025a] employs explicit, rule-based accuracy rewards for mathematics, as well as compiler- or test-based rewards for coding tasks. This approach demonstrates that large-scale reinforcement learning, specifically, Group Relative Policy Optimization (GRPO), can induce sophisticated reasoning behaviors even in base models prior to subsequent alignment stages.

This shift reframes reasoning as a capability that can be explicitly trained and scaled [OpenAI, 2025a,b]: LRMs allocate significant test-time compute to generate, evaluate, and revise intermediate chain-of-thought [Wei et al., 2022], and their performance rises as this compute budget increases. This dynamic introduces a complementary path to capability gains, orthogonal to data and parameter scaling during pre-training [Aghajanyan et al., 2023, Kaplan et al., 2020], while leveraging a reward maximization objective [Silver et al., 2021], automatically checkable rewards wherever reliable verifiers exist (e.g., competition mathematics [Guo et al., 2025a, Jaech et al., 2024], competitive programming [El-Kishky et al., 2025], and selected scientific domains [Bai et al., 2025]). Furthermore, RL can overcome data limitations [Shumailov et al., 2024, Villalobos et al., 2022] by enabling self-generated training data [Silver et al., 2018, Zhao et al., 2025a]. As a result, RL is increasingly regarded as a promising technology for achieving Artificial SuperIntelligence (ASI) on a broader range of tasks through continual scaling.

At the same time, further scaling of RL for LRMs introduces new constraints, not only in computational resources, but also in algorithm design, training data, and infrastructure. How and where RL for LRMs can be scaled to achieve high-level intelligence and generate real-world value remain unresolved issues. Therefore, we argue that it is timely to revisit the development of this domain and explore strategies to enhance the scalability of RL toward artificial superintelligence. In summary, this survey reviews recent work on RL for LRMs as follows:

- We introduce the preliminary definitions of RL modeling in the context of LRMs (§ 2.1) and outline the development of frontier reasoning models since the release of OpenAI o1 (§ 2.2).

- We review recent literature on the foundational components of RL for LRMs, including reward
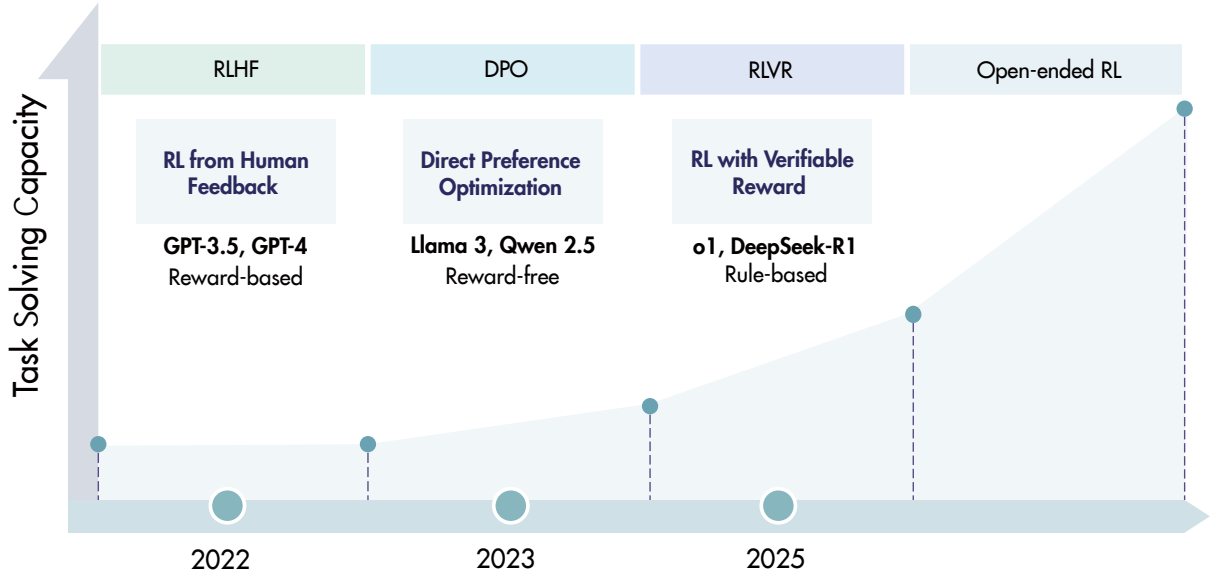
Figure 2 | RLHF and DPO have been the two predominant RL methodologies for human alignment in recent years. In contrast, RLVR represents an emerging trend in RL for LRMs, significantly enhancing their capacity for complex task solving. The next stage of scaling RL for LLMs remains an open question, with open-ended RL presenting a particularly challenging and promising direction.

    design (§ 3.1), policy optimization (§ 3.2), and sampling strategies (§ 3.3), comparing the different research directions and technical approaches for each component.

- We discuss foundational and still controversial problems in RL for LRMs (§ 4), such as the role of RL (§ 4.1), RL versus Supervised Fine-Tuning (SFT) (§ 4.2), model priors (§ 4.3), training recipes (§ 4.4), and reward definitions (§ 4.5). We argue that these issues warrant further exploration to enable continued scaling of RL.

- We examine training resources for RL (§ 5), including static corpora (§ 5.1), dynamic environments (§ 5.2), and training infrastructure (§ 5.3). While these resources are reusable in both research and production, further standardization and development are needed.

- We review applications of RL to a wide range of tasks (§ 6), such as coding tasks (§ 6.1), agentic tasks (§ 6.2), multimodal tasks (§ 6.3), multi-agent systems (§ 6.4), robotics tasks (§ 6.5), and medical applications (§ 6.6).

- Finally, we discuss future directions in RL for language models (§ 7), covering novel algorithms, mechanisms, features, and additional research avenues.

2. Preliminaries

2.1. Background

In this subsection, we introduce the basic components of RL and describe how language models can be configured as agents within RL frameworks. As shown in Figure 3, RL provides a general framework for sequential decision making, in which an agent interacts with an environment by taking actions to maximize cumulative reward. In classical RL, the problem is typically formulated as a Markov
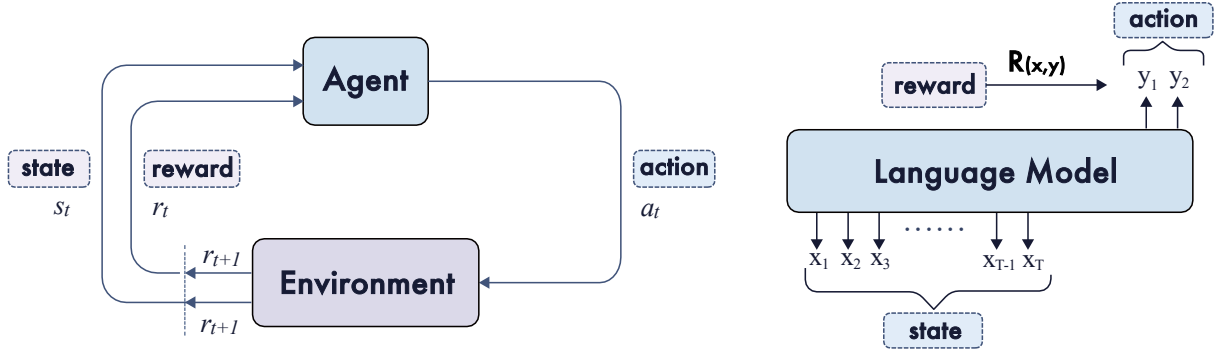
Figure 3 | Basic components of RL and language models (LMs) as agents. The agent selects actions, while the environment provides states and rewards at each turn. In the context of LMs, completion tokens are treated as actions, which are concatenated with the context to form the state. Rewards are typically assigned at the level of the entire response.

Decision Process (MDP) [Sutton et al., 1998], which is defined by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, R, \gamma)$. The main components include a state space $\mathcal{S}$, an action space $\mathcal{A}$, transition dynamics $\mathcal{P} : \mathcal{S} \times \mathcal{A} \mapsto \mathcal{S}$, a reward function $R : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$, and a discount factor $\gamma \in [0, 1]$. At each step, the agent observes a state s_t, selects an action a_t according to its policy π_θ parameterized by θ, receives a reward r_t, and transits to the next state s_{t+1}. When applying RL to language models, these concepts can be naturally mapped to the language domain with minimal adaptation. The mapping is summarized as follows:

- **Prompt/Task (x):** Corresponds to the initial state or environment context, drawn from a data distribution and corresponding to the dataset $\mathcal{D}$.

- **Policy (π_θ):** Represents the language model, which generates a sequence of tokens $y = (a_1, \ldots, a_T)$ in response to the prompt, where $T = |y|$ denotes the sequence length.

- **State (s_t):** Defined as the prompt together with the sequence generated so far, i.e., $s_t = (x, a_{1:t-1})$.

- **Action (a_t):** The token chosen at step t from the vocabulary $\mathcal{V}$, which is also $\mathcal{A}$.

- **Transition Dynamics ($\mathcal{P}$):** The state transition is usually deterministic in the context of LLM since $s_{t+1} = [s_t, a_t]$, where $[\cdot, \cdot]$ denotes string concatenation.

- **Reward ($R(x, y)$ or r_t):** Typically assigned at the end of the sequence (sequence-level), denoted $R(x, y)$, but may also be decomposed into token-level rewards r_t with process supervision.

- **Return (G):** The cumulative (optionally discounted) reward accrued over the whole trajectory y for prompt x. With sequence-level feedback it reduces to the single scalar $R(x, y)$; with token-level feedback it aggregates the per-token rewards r_t (typically with γ=1 for finite T).

In this setting, the learning objective [Sutton et al., 1998] is to maximize the expected reward over the data distribution $\mathcal{D}$, that is,

$$\max_\theta \mathcal{J}(\theta) := \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta(x)}[R(x, y)]. \tag{1}$$

In practice, it is common to regularize the learned policy towards a reference policy π_{ref}, often implemented as KL-divergence constraints to stabilize training and maintain language quality. In the following sections, we present various algorithms that build upon this fundamental formulation.

2.2. Frontier Models

In this subsection, we provide an overview of state-of-the-art large reasoning models trained with RL-like methods, organized roughly chronologically along three major directions: LRMs, agentic LRMs, and multimodal LRMs.

Over the past year, RL has progressively expanded the frontier of reasoning models and their applications. The first large reasoning models, OpenAI's o1 [Jaech et al., 2024] series, established the effectiveness of scaling both train-time RL and test-time compute towards more powerful reasoning abilities, achieving leading results on mathematics, coding, and science benchmarks. DeepSeek's flagship model R1 [Guo et al., 2025a] followed as the first open-source model to match o1's performance across benchmarks. It employs a multi-stage training pipeline to ensure well-rounded model abilities, and explores the route of pure RL without supervised finetuning (i.e., Zero RL). Other proprietary model releases promptly followed: Claude-3.7-Sonnet [Anthropic, 2025a] featured hybrid reasoning, Gemini 2.0 and 2.5 [Comanici et al., 2025] introduced longer context lengths, Seed-Thinking 1.5 [Seed et al., 2025b] featured generalization across domains, and the o3 [OpenAI, 2025b] series showcased increasingly advanced reasoning abilities. Recently, OpenAI introduced their first open-source reasoning model gpt-oss-120b [Agarwal et al., 2025a], and subsequently GPT5 [OpenAI, 2025a], their most capable AI system to date, which flexibly switches between an efficient model and a deeper reasoning model GPT-5 thinking. Parallel open-source efforts continued to expand the landscape. Within the Qwen family, QwQ-32B [Team, 2025g] matched R1's performance, and was followed by the Qwen3 [Yang et al., 2025a] series, with the representative model Qwen3-235B further improving benchmark scores. The Skywork-OR1 [He et al., 2025d] suite of models were based on R1-distilled models, and achieved scalable RL training through effective data mixtures and algorithmic innovations. Minimax-M1 [Chen et al., 2025a] was the first model to introduce hybrid attention to scale RL efficiently. Other works include Llama-Nemotron-Ultra [Bercovich et al., 2025], which aimed to balance accuracy and efficiency; Magistral 24B [Rastogi et al., 2025], trained through RL from scratch without distillation from prior models; and Seed-OSS [Team, 2025a], emphasizing long-context reasoning abilities.

Model reasoning improvements have in turn extended their use cases in coding and agentic scenarios. The Claude series has been known for their leading performance on agentic coding tasks, and this was exemplified by Claude-4.1-Opus [Anthropic, 2025b], which further pushed forward the state-of-the-art results on SWE-bench [Jimenez et al., 2023]. Kimi K2 [Team, 2025d] is a recent representative agentic model which was specifically optimized for agentic tasks, forging large-scale agentic training data synthesis and a general RL procedure that accommodates non-verifiable rewards. Shortly after, both the GLM4.5 [Zeng et al., 2025a] and DeepSeek-V3.1 releases emphasized tool-use and agentic tasks, showing substantial improvements on relevant benchmarks.

Multimodality is a key component behind the widespread adoption of reasoning models. Most frontier proprietary models, including GPT-5, o3, Claude, and Gemini families, are natively multimodal. Gemini-2.5 [Comanici et al., 2025] notably emphasized strong performance across text, images, video, and audio. On the open-source side, Kimi 1.5 [Team, 2025d] represents an early effort towards multimodal reasoning, highlighting long context scaling as well as joint reasoning over text and vision domains. QVQ [Qwen Team, 2025] excels in visual reasoning and analytical thinking, while Skywork R1V2 [Wang et al., 2025j] balances reasoning and general abilities through hybrid RL, using both MPO and GRPO. As notable additions to the InternVL series, InternVL3 [Zhu et al., 2025c] adopted a unified native multimodal pretraining phase, and later InternVL3.5 [Wang et al., 2025n] used a two-stage cascade RL framework, achieving improved efficiency and versatility. More recently, the Intern-S1 [Bai et al., 2025] model focused on multimodal scientific reasoning across diverse domains, benefiting from a mixture-of-rewards design during online RL to facilitate simultaneous training on a
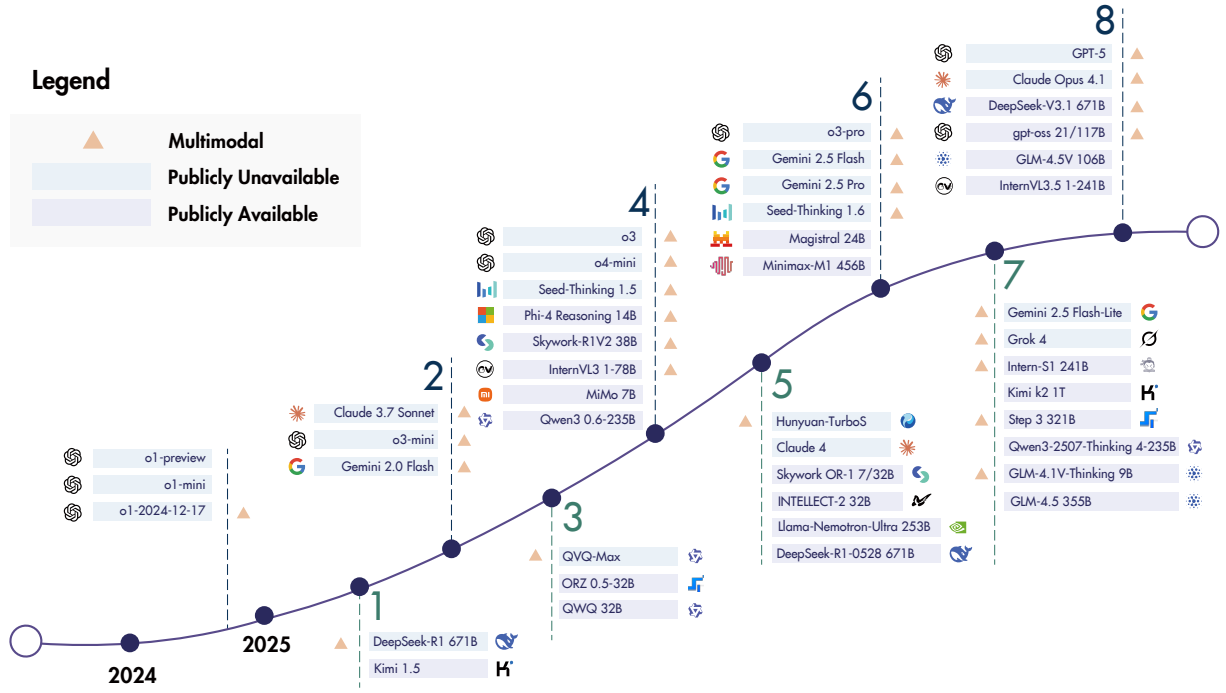
Figure 4 | Timeline of representative open-source and closed-source reasoning models trained with RL, including language models, multimodal models, and agentic models.

wide range of tasks. Other recent models include Step3 [Wang et al., 2025a], designed for efficient training and minimizing decoding costs, and GLM-4.5V [Team et al., 2025a], with state-of-the-art performance across most visual multimodal benchmarks.

In addition to the aforementioned models, we provide a comprehensive list of reasoning models in Figure 4 and detailed information on open-source models in Table 1.

2.3. Related Surveys

In this subsection, we compare recent surveys related to RL and LLMs. Several surveys focus primarily on RL itself, covering both classical RL and its recent extensions. Ghasemi et al. [2024] present a general RL survey covering algorithms and real-world challenges, Huh and Mohapatra [2023] focus on multi-agent RL, Zhang et al. [2024b] review self-play techniques, and Wu et al. [2025h] survey RL in computer vision tasks. While these works offer broad perspectives on RL, they do not explicitly address its application to LLMs. In contrast, other surveys center on LLMs and their emerging capabilities, such as long chain-of-thought reasoning [Chen et al., 2025m, Li et al., 2025u, Xia et al., 2024] and adaptive behaviors [Feng et al., 2025c, Sui et al., 2025], where RL is often introduced as a key method to support these advances. Zhao et al. [2023a] provide a broad overview of LLM architectures and applications, while more recent works concentrate specifically on reasoning abilities. Zhang et al. [2025a] survey replication studies on reasoning LLMs in the wake of DeepSeek-R1, Chen et al. [2025m] examine long chain-of-thought reasoning, and Li et al. [2025u] analyze the transition from System 1 to System 2 reasoning. These studies highlight RL-based methods such as RLHF and RLVR as useful tools, but treat them as only one element among a wide range of reasoning strategies. Sun et al. [2025b] offer a broader, structured take on reasoning via foundation models. It highlights key foundation models that are either proposed or adapted specifically for reasoning, as well as recent progress across diverse reasoning tasks, methodologies, and benchmarks. Zhang et al. [2025b]

Table 1 | Comparison of representative open-source models trained with RL. OPMD denotes Online Policy Mirror Descent; MPO denotes Mixed Preference Optimization; CISPO denotes Clipped IS-weight Policy Optimization. T, I, and V indicate Text, Image, and Video modalities, respectively.

Date	Model	Organization	Architecture	Parameters	Algorithm	Modal	Link
2025.01	DeepSeek-R1 [Guo et al., 2025a]	DeepSeek	MoE/MLA	671B	GRPO	Text	〇 🤗
2025.03	ORZ [Hu et al., 2025b]	StepAI	Dense	0.5-32B	PPO	Text	〇 🤗
2025.03	QwQ [Team, 2025g]	Alibaba Qwen	Dense	32B	-	Text	〇 🤗
2025.04	Phi-4 Reasoning [Abdin et al., 2025]	Microsoft	Dense	14B	GRPO	Text	〇 🤗
2025.04	Skywork-R1V2 [Wang et al., 2025j]	Skywork	Dense	38B	MPO/GRPO	T/I	〇 🤗
2025.04	InternVL3 [Zhu et al., 2025c]	Shanghai AI Lab	Dense	1-78B	MPO	T/I/V	〇 🤗
2025.04	MiMo [Xiaomi et al., 2025]	Xiaomi	Dense	7B	GRPO	Text	〇 🤗
2025.04	Qwen3 [Yang et al., 2025a]	Alibaba Qwen	MoE/Dense	0.6-235B	GRPO	Text	〇 🤗
2025.05	Llama-Nemotron-Ultra [Bercovich et al., 2025]	NVIDIA	Dense	253B	GRPO	Text	〇 🤗
2025.05	INTELLECT-2 [Team et al., 2025b]	Intellect AI	Dense	32B	GRPO	Text	🤗
2025.05	Hunyuan-TurboS [Team et al., 2025c]	Tencent	Hybrid MoE	560B	GRPO	Text	〇 🤗
2025.05	Skywork OR-1 [He et al., 2025d]	Skywork	Dense	7B/32B	GRPO	Text	〇 🤗
2025.05	DeepSeek-R1-0528 [Guo et al., 2025a]	DeepSeek	MoE/MLA	671B	GRPO	Text	〇 🤗
2025.06	Magistral [Rastogi et al., 2025]	Mistral AI	Dense	24B	GRPO	Text	🤗
2025.06	Minimax-M1 [Chen et al., 2025a]	Minimax	Hybrid MoE	456B	CISPO	Text	〇 🤗
2025.07	Intern-S1 [Bai et al., 2025]	Shanghai AI Lab	MoE	241B	GRPO	T/I/V	〇 🤗
2025.07	Kimi K2 [Team, 2025c]	Kimi	MoE	1T	OPMD	Text	〇 🤗
2025.07	Step 3 [Wang et al., 2025a]	Step AI	MoE	321B	-	T/I/V	〇 🤗
2025.07	Qwen3-2507 [Yang et al., 2025a]	Alibaba Qwen	MoE/Dense	4-235B	GSPO	Text	〇 🤗
2025.07	GLM-4.1V-Thinking [Team et al., 2025a]	Zhipu AI	Dense	9B	GRPO	T/I/V	〇 🤗
2025.07	GLM-4.5 [Zeng et al., 2025a]	Zhipu AI	MoE	355B	GRPO	Text	〇 🤗
2025.07	Skywork-R1V3 [Shen et al., 2025b]	Skywork	Dense	38B	GRPO	T/I	〇 🤗
2025.08	gpt-oss [Agarwal et al., 2025a]	OpenAI	MoE	117B/21B	-	Text	〇 🤗
2025.08	Seed-OSS [Team, 2025a]	Bytedance Seed	Dense	36B	-	Text	〇 🤗
2025.08	GLM-4.5V [Team et al., 2025a]	Zhipu AI	MoE	106B	GRPO	T/I/V	〇 🤗
2025.08	InternVL3.5 [Wang et al., 2025n]	Shanghai AI Lab	MoE/Dense	1-241B	MPO/GSPO	T/I/V	〇 🤗
2025.09	ERNIE-4.5-Thinking [Baidu-ERNIE-Team, 2025]	Baidu	MoE	21B-A3B	-	Text	🤗

examine how RL can endow LLMs with autonomous decision-making and adaptive agentic capabilities. Xu et al. [2025a] move closer to our focus by discussing reinforced reasoning for LLMs, emphasizing how trial-and-error optimization can improve complex reasoning. Wu [2025] complement this view by surveying reward models and strategies for learning from feedback. Nevertheless, these works remain oriented towards reasoning performance or reward design, rather than offering a systematic treatment of RL methods as a whole for LLMs. Srivastava and Aggarwal [2025] represent a more recent attempt to bridge the two fields by reviewing RL algorithms for LLM alignment and enhancement, primarily through methods such as RLHF [Christiano et al., 2017], RLAIF [Lee et al., 2024b], and DPO [Rafailov et al., 2023]. It remains primarily focused on alignment rather than reasoning capabilities.

Unlike previous surveys that cover either general RL or reasoning in LLMs, we place RL at the center and provide a systematic synthesis of its role throughout the LLM training lifecycle, including reward design, policy optimization, and sampling strategies. Our aim is to identify new directions for scaling reinforcement learning in LRMs toward ASI, focusing on long-term interactions and evolution.

3. Foundational Components

In this section, we review the foundational components of RL for LRMs, including reward design (§ 3.1), policy optimization algorithms (§ 3.2), and sampling strategies (§ 3.3). The taxonomy of the foundational components are shown in Figure 5.

3.1. Reward Design

In this subsection, we provide a comprehensive examination of reward design in RL for LRMs. We begin in § 3.1.1 with verifiable rewards, which offer a natural starting point. There are substantial advances in this direction, exemplified by the success of DeepSeek-R1, which demonstrated the scalability of RL through verifiable reward mechanisms. In contrast, § 3.1.2 examines generative rewards, wherein the model is engaged to either verify or directly generate reward signals. However, both verifiable and generative rewards are typically expressed as sparse numerical feedback. An important complementary dimension lies in the density of the reward signal. § 3.1.3 accordingly examines approaches that incorporate dense rewards. A further axis of categorization pertains to whether rewards are computed from external ground truth or instead estimated directly by the model. This distinction motivates our discussion of unsupervised rewards in § 3.1.4. Building upon these four categories, we then turn in § 3.1.5 to reward shaping, where we analyze strategies for combining or transforming diverse reward signals to facilitate learning.

3.1.1. Verifiable Rewards

Takeaways

- Rule-based rewards provide scalable and reliable training signals for RL, especially in math and code tasks, by leveraging accuracy and format checks.

- Verifier's law highlights that tasks with clear and automatic verification enable efficient RL optimization, while subjective tasks remain challenging.

Rule-based Rewards. The reward serves as the training signal of RL, determining the optimization direction [Guo et al., 2025a]. Recently, rule-based verifiable rewards have been predominantly employed to train LRMs in large-scale RL. Such rewards enable the reliable enhancement of mathematical and coding reasoning abilities by encouraging longer and more reflective chain-of-thought [Guo et al.,

Foundational Components § 3

Reward Design § 3.1

- **Generative Rewards § 3.1.2**
  - *Model-based Verifiers for Verifiable Tasks*: e.g., TinyV [Xu et al., 2025g]; xVerify [Chen et al., 2025b]; CompassVerifier [Liu et al., 2025m]; General Reasoner [Ma et al., 2025c]; Seed-Thinking-Verifier [Seed et al., 2025a]
  - *Reasoning Reward Models*: e.g., AIR [He et al., 2025a]; DeepSeek-GRM [Liu et al., 2025w]; Pairwise-RL [Xu et al., 2025c]; J1 [Whitehouse et al., 2025]; RM-R1 [Chen et al., 2025p]; RRM [Guo et al., 2025b]; Think-RM [Hong et al., 2025b]; GRAM [Wang et al., 2025c]; OMNI-THINKER [Li et al., 2025b]; LI-BRA [Zhou et al., 2025c]; TP-GRPO [He et al., 2025f]; CAPO [Xie et al., 2025b]
  - *Rubric-based Rewards*: e.g., RaR [Gunjal et al., 2025]; Rubicon [Huang et al., 2025f]; RLCF [Viswanathan et al., 2025]; Writing-Zero [Jia et al., 2025]; ProxyReward [Guo et al., 2025e]; ReviewRL [Zeng et al., 2025c]; RuscaRL [Zhou et al., 2025f]
  - *Co-Evolving Systems*: e.g., self rewarding GRM [Yuan et al., 2024]; RL Tango [Zha et al., 2025]; PAG [Jiang et al., 2025e]; URPO [Lu et al., 2025e]; PCL [Fei et al., 2025b]; K2 [Team, 2025c]; Cooper [Hong et al., 2025a]; Critique-GRPO [Zhang et al., 2025l]

- **Dense Rewards § 3.1.3**
  - *Token-level*: e.g., Implicit PRM [Yuan et al., 2025d]; PRIME [Cui et al., 2025a]; SRPO [Fei et al., 2025a]; Entropy Advantage [Cheng et al., 2025a]; HICRA [Wang et al., 2025g]
  - *Step-level*: e.g., PURE [Cheng et al., 2025b]; Tango [Zha et al., 2025]; VinePPO [Kazemnejad et al., 2025]; SPO [Guo et al., 2025c]; TreeRPO [Yang et al., 2025f]; TreeRL [Hou et al., 2025]; TreePO [Li et al., 2025q]
  - *Turn-level*: e.g., TLCA [Zeng et al., 2025d]; LLM-RD [Lee et al., 2025]; ToolRL [Qian et al., 2025]; MUA-RL [Zhao et al., 2025d]; ESDP [Zhu et al., 2024]; SWEET-RL [Zhou et al., 2025g]; GELI [Lee et al., 2024a]; SPA-RL [Wang et al., 2025e]

- **Unsupervised Rewards § 3.1.4**
  - *Model-Specific*: e.g., TTRL [Zuo et al., 2025b]; ETTRL [Liu et al., 2025c]; EMPO [Zhang et al., 2025h]; Spurious Rewards [Shao et al., 2025]; Absolute Zero [Zhao et al., 2025a]; RLIF [Zhao et al., 2025e]; R-Zero [Huang et al., 2025a]; SRT Shafayat et al. [2025]; EM-RL [Agarwal et al., 2025b]; SeRL [Fang et al., 2025a]; RLSC [Li et al., 2025g]; Co-Reward [Zhang et al., 2025v]; CoVo [Zhang et al., 2025g]; CAGSR [Kiruluta et al., 2025]; RENT [Prabhudesai et al., 2025]; RLSF [van Niekerk et al., 2025]; SSR-Zero [Yang et al., 2025e]; MINIMO [Poesia et al., 2024]; SQLM [Chen et al., 2025i]
  - *Model-Agnostic*: e.g., SEAL [Zweiger et al., 2025]; RPT [Dong et al., 2025c]; Xin et al. [2025]

- **Rewards Shaping § 3.1.5**
  - *Rule-based Reward Shaping*: e.g., Qwen2.5-Math [Yang et al., 2024a]; DeepSeek-R1 [Guo et al., 2025a]; Laser [Liu et al., 2025o]
  - *Structure-based Reward Shaping*: e.g., GRPO [Shao et al., 2024]; RLOO [Ahmadian et al., 2024]; PKPO [Walder and Karkhanis, 2025]; Pass@k Training [Chen et al., 2025v]

Policy Algorithm § 3.2

- **Critic-based Algorithms § 3.2.2**
  - e.g., PPO [Schulman et al., 2017a]; VCPPO [Yuan et al., 2025f]; VAPO [Yue et al., 2025c]; PRIME [Cui et al., 2025a]
- **Critic-Free Algorithms § 3.2.3**
  - e.g., REINFORCE [Cheng et al., 2025d]; ReMax [Li et al., 2023c]; RLOO [Cheng et al., 2025d]; REINFORCE++ [Hu, 2025]; GRPO [Shao et al., 2024]; DAPO [Yu et al., 2025d]; CISPO [Chen et al., 2025a]; Dr.GRPO [Liu et al., 2025t]; GSPO [Zheng et al., 2025a]; VinePPO [Kazemnejad et al., 2025]; MDPO [Team, 2025d]; LitePPO [Liu et al., 2025v]
- **Regularization Objectives § 3.2.5**
  - *KL Regularization*: e.g., SimpleRL [Zeng et al., 2025e]; ProRL [Liu et al., 2025i]; OREAL [Lyu et al., 2025]; Phi4-reasoning [Abdin et al., 2025]; Kimi-K1.5 [Team, 2025d], Archer [Wang et al., 2025h]
  - *Entropy Regularization*: e.g., DAPO [Yu et al., 2025d]; KL-Cov/Clip-Cov [Cui et al., 2025b]; HighEntropy-RL [Wang et al., 2025l];
  - *Length Penalty*: e.g., ALP [Xiang et al., 2025]; LASER [Liu et al., 2025o]; L1 [Aggarwal and Welleck, 2025]; O1-pruner [Luo et al., 2025a]
- **Off-Policy Optimization § 3.2.4**
  - *Off-policy*: e.g., SPO [Cohen et al., 2025]; TOPR [Roux et al., 2025]; ReMix [Liang et al., 2025a]
  - *Mix-policy*: e.g., HPT [Lv et al., 2025]; LUFFY [Yan et al., 2025a]; Re-LIFT [Ma et al., 2025a]; UFT [Liu et al., 2025j]; BREAD [Zhang et al., 2025o]; SRFT [Fu et al., 2025c]; Prefix-RFT [Huang et al., 2025g]

Sampling Strategy § 3.3

- **Dynamic and Structured Sampling § 3.3.1**
  - *Dynamic Sampling*: e.g., PRIME [Cui et al., 2025a]; DAPO [Yu et al., 2025d]; K1.5 [Team, 2025d]; SEC [Chen et al., 2025o]; E2H [Parashar et al., 2025]; AdaRFT [Shi et al., 2025b]; SPaRFT [Do et al., 2025]; ARPO [Dong et al., 2025b]; DARS [Yang et al., 2025g];
- **Hyper-Parameters Adjustment § 3.3.2**
  - e.g., DeepScaleR [Luo et al., 2025c]; E3-RL4LLMs [Liao et al., 2025b]; Pro-RL [Liu et al., 2025i]; AceReason-Nemotron [Chen et al., 2025q]; T-PPO [Fan et al., 2025b]; POLARIS [An et al., 2025]; Confucius3-Math [Wu et al., 2025e]; GFPO [Shrivastava et al., 2025]
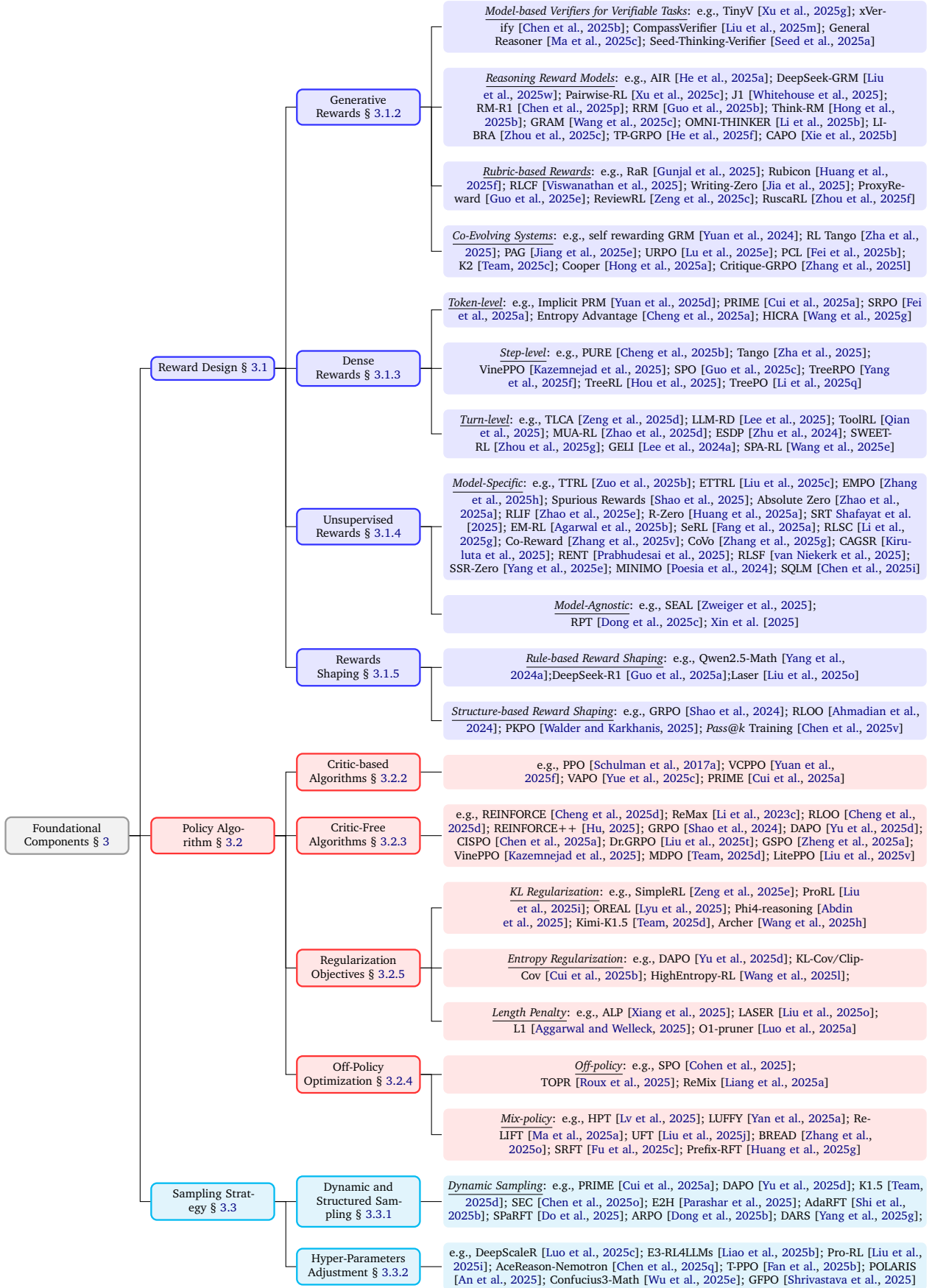
Figure 5 | Taxonomy of foundational components and representative works for each direction.

2025a, Team, 2025c, Yu et al., 2025d]. This paradigm was formalized as RLVR in the Tülu 3 [Lambert et al., 2024], which replaces a learned reward model with a programmatic verifier (e.g., answer checkers or unit tests). Such verifiers provide binary, checkable signals in domains with objectively verifiable outcomes. Similar rule-based approaches to verifiable reward design were subsequently integrated into DeepSeek's training pipeline. For instance, DeepSeek-V3 [Liu et al., 2024] explicitly incorporated a rule-based reward system tailored to deterministic tasks, while DeepSeek-R1 [Guo et al., 2025a] further employed accuracy-based and format-based rewards. Rule-based rewards stand in contrast to outcome-based or process-based Reward Models (RMs), such as standard RLHF with a learned reward model trained on human preference rankings [Ouyang et al., 2022] and Process Reward Models (PRMs) trained on step-level annotations [Setlur et al., 2024, Sun et al., 2025c, Yuan et al., 2025d]. DeepSeek-V3 and DeepSeek R1 demonstrate that RMs may suffer from reward hacking when scaled to large-scale RL settings, but by leveraging rule-based rewards wherever possible, we ensure greater reliability by making the system resistant to manipulation and exploitation [Guo et al., 2025a, Liu et al., 2024]. In practice, two kinds of rule-based verifiable rewards are widely used:

- **Accuracy rewards:** For tasks with deterministic outcomes (e.g., math), the policy must produce the final solution within a prescribed delimiter (commonly \boxed{...}). An automatic checker then compares this output to the ground truth. For coding tasks, unit tests, or compilers provide the pass/fail signal [Albalak et al., 2025, Chen et al., 2025r, Guo et al., 2025a].

- **Format rewards:** These impose a structural constraint requiring the model to place its private chain-of-thought between <think> and </think>, and to output the final answer in a separate field (e.g., <answer>...</answer>). This improves reliable parsing and verification in large-scale RL [Guo et al., 2025a, Lambert et al., 2024].

Rule-based Verifier. Rule-based rewards are typically derived from rule-based verifiers. These rely on a large collection of manually written equivalence rules to determine whether a predicted answer matches the ground truth. Currently, widely used mathematical verifiers are primarily built on the Python libraries Math-Verify[1] and SymPy[2]. In addition, some works such as DAPO [Yu et al., 2025d] and DeepScaleR [Luo et al., 2025c], also provide open-source and well-established verifiers. Recently, Huang et al. [2025e] highlight the distinctive limitations associated with both rule-based and model-based verifiers, to inform the design of more reliable reward systems.

In practice, tasks such as mathematical problem solving and code generation are difficult to solve yet comparatively easy to verify, thereby satisfying the main criteria for efficient RL optimization [Guo et al., 2025a, He et al., 2025d]: the existence of clear ground truth, the availability of rapid automated verification, the scalability of evaluating many candidate solutions, and a reward signal that is closely aligned with correctness. By contrast, tasks lacking fast or objective verification (e.g., open-ended question answering or free-form writing) remain challenging for outcome-based RL, as they rely on noisy learned reward models or subjective human feedback [Yu et al., 2025e, Zhou et al., 2025e]. Verifier's Law posits that the ease of training AI systems to perform a task is proportional to the degree to which the task is verifiable[3]. It emphasizes that once a task can be equipped with robust automated feedback, it becomes amenable to rapid improvement via RL. The successful applications discussed in §6 substantiate this principle, as their central challenge lies in the design of reliable verifiable feedback. Conversely, many of the open problems highlighted in §7 arise precisely from the absence of dependable automated rewards.

---

[1] https://github.com/huggingface/Math-Verify
[2] https://www.sympy.org/
[3] https://www.jasonwei.net/blog/asymmetry-of-verification-and-verifiers-law

3.1.2. Generative Rewards

> **Takeaways**
>
> - Generative Reward Models (GenRMs) extend RL to subjective, non-verifiable domains by providing nuanced, text-based feedback, overcoming the limitations of rule-based systems.
>
> - A dominant trend is training RMs to reason before judging, often using structured rubrics to guide evaluation or co-evolving them with the policy model in a unified RL loop.

While rule-based rewards provide reliable signals for verifiable tasks, as discussed previously (§ 3.1.1), their applicability is limited. Many complex reasoning tasks, particularly in open-ended or creative domains, lack objective ground truth, making them intractable for simple verifiers. To bridge this gap, GenRMs have emerged as a powerful alternative. Instead of outputting a simple scalar score, GenRMs leverages the generative capabilities of LRMs to produce structured critiques, rationales, and preferences, providing a more interpretable and nuanced reward signal [Mahan et al., 2024, Zhang et al., 2024a]. This approach addresses two key challenges: first, it improves the robustness of verification for verifiable tasks that are difficult to parse; second, and more importantly, it enables the application of RL to subjective, non-verifiable domains.

Model-based Verifiers for Verifiable Tasks. A primary challenge with rule-based systems is their brittleness; they often produce false negatives when a model generates a correct answer in an unexpected format. To mitigate this, one line of research uses *Specification-Based GenRMs* as flexible, model-based verifiers. These models are trained to semantically assess the equivalence between a model's free-form output and a reference answer. This approach has been used to develop lightweight verifiers that augment existing rule-based systems [Xu et al., 2025g], as well as more comprehensive, multi-domain verifiers capable of handling diverse data types and reasoning tasks [Chen et al., 2025b, Liu et al., 2025m, Ma et al., 2025c, Seed et al., 2025a]. By replacing or supplementing rigid string matching with learned semantic judgment, these verifiers provide more accurate reward signals for RL in verifiable domains.

Generative Rewards for Non-Verifiable Tasks. Another core application of GenRMs is *Assessment-Based GenRMs*, which enable RL for tasks where Verifier's Law does not hold. This paradigm has evolved from using powerful LLMs as zero-shot evaluators to sophisticated, co-evolving systems. We can categorize these approaches based on their core design principles.

- **Reasoning Reward Models (Learning to Think)**: A major advancement beyond simple preference prediction is to train RMs to explicitly reason before rendering a judgment. This approach, foundational to the LLM-as-a-Judge concept [Li et al., 2023b, Zheng et al., 2023], involves prompting the RM to generate a CoT critique or rationale. For instance, CLoud RMs first generate a natural language critique and then use it to predict a scalar reward [Ankner et al., 2024]. This principle of formulating reward modeling as a reasoning task is now central to state-of-the-art RMs, which are trained to produce detailed rationales before assigning a score or preference [Chen et al., 2025p, Guo et al., 2025b, Hong et al., 2025b, Liu et al., 2025w, Wang et al., 2025c, Zhou et al., 2025c]. To further improve their judgment capabilities, these reasoning RMs are often trained with RL themselves, using simple, verifiable meta-rewards based on the correctness of their final verdict [Chen et al., 2025l, Whitehouse et al., 2025]. This line of work also explores different reward formats, such as deriving soft rewards from token probabilities [Mahan et al., 2024, Su et al., 2025c, Zhang et al., 2024a] and weighing the trade-offs between pointwise and pairwise scoring schemes [He et al., 2025a, Xu et al., 2025c].

- **Rubric-based Rewards (Structuring Subjectivity)**: To anchor the evaluation of subjective tasks

in more consistent criteria, many frameworks employ structured rubrics. Unlike rule-based approaches that rely on hard-coded logic for objective, verifiable tasks, rubric-based methods leverage natural language descriptions to capture nuanced evaluation criteria for subjective, non-verifiable domains where traditional binary rules would be insufficient. This involves using an LLM to either generate or follow a checklist of principles to guide its assessment. Frameworks like RaR [Gunjal et al., 2025], Rubicon [Huang et al., 2025f], and RLCF [Viswanathan et al., 2025] use such rubrics to produce fine-grained, multi-faceted rewards. This concept extends to decomposing high-level tasks into a set of verifiable proxy questions [Guo et al., 2025e] or generating domain-specific principles, such as for creative writing [Jia et al., 2025] or scientific reviews [Zeng et al., 2025c]. Furthermore, rubrics can serve a dual purpose as both instructional scaffolds to guide policy exploration and as criteria for the final reward [Zhou et al., 2025f].

- **Co-Evolving Systems (Unifying Policy and Reward):** The most advanced paradigm moves beyond a static policy-reward relationship and toward dynamic systems where the generator and verifier improve together. This can occur through:

  - **Self-Rewarding**, where a single model generates its own training signals. This was notably demonstrated in Self-Rewarding Language Models [Yuan et al., 2024] and has been operationalized in frameworks where a model alternates between policy and verifier roles [Jiang et al., 2025e], performs self-correction based on its own critique [Team, 2025c, Xiong et al., 2025b, Zhang et al., 2025l], or internalizes the reward function via post-completion learning [Fei et al., 2025b].

  - **Co-Optimization**, where the policy and a separate reward model are trained concurrently. For example, RL Tango jointly trains the generator and a process-level GenRM using a shared outcome-level reward [Zha et al., 2025]. Similarly, Cooper co-optimizes both models to enhance robustness and mitigate reward hacking [Hong et al., 2025a]. Other works unify the policy ("player") and reward ("referee") functions within a single model trained via a unified RL loop [Lu et al., 2025e].

This evolution from static judges to dynamic, co-evolving systems is often supported by hybrid reward schemes that combine rule-based and generative signals [Li et al., 2025b, Seed et al., 2025a]. Additionally, GenRMs are being adapted to provide more granular, process-level feedback to address the credit assignment problem in complex reasoning chains [He et al., 2025f, Xie et al., 2025b, Zhao et al., 2025b]. In essence, generative rewards are proving indispensable for scaling RL to the full spectrum of tasks targeted by general-purpose LRMs.

3.1.3. *Dense Rewards*

> **Takeaways**
>
> - Dense rewards (e.g., process reward models) provide fine-grained credit assignment and improve training efficiency and optimization stability in RL.
>
> - Scaling remains challenging for tasks like open-domain text generation due to the difficulty of defining dense rewards or using verifiers.

In classical RL such as gaming and robotic manipulation tasks [Liu et al., 2022, Schrittwieser et al., 2020, Sun et al., 2025d], dense rewards provide frequent feedback at (nearly) every decision step. Such shaping shortens the credit assignment horizon and often improves sample efficiency and optimization stability, but it also risks mis-specification and reward hacking if the signal is poorly

Table 2 | Definitions of action and reward granularity in RL for language models ($z^{(u)}$ is the environment feedback at turn u).

Granularity	Action	Reward	Return (G)
Trajectory	Entire sequence $y = (a_1, \ldots, a_T)$	Scalar $R(x, y)$	$R(x, y)$
Token	Each token $a_t \in \mathcal{V}$	$r_t = R(x, a_{1:t})$	$\sum_{t=1}^{T} \gamma^{t-1} r_t$
Step	Segment $y^{(k)}$ (e.g., sentence)	$r_k = R(x, y^{(1:k)})$	$\sum_{k=1}^{K} \gamma^{k-1} r_k$
Turn (Agent)	Agent response $y^{(u)}$ per turn	$r_u = R(x, y^{(1:u)}, z^{(1:u)})$	$\sum_{u=1}^{U} \gamma^{u-1} r_u$

designed [Hadfield-Menell et al., 2017]. As for LLM reasoning, dense rewards are typically process-based signals that supervise intermediate steps rather than only outcomes, and they have been found effective, often outperforming outcome-based rewards [Lightman et al., 2024, Uesato et al., 2022]. Based on the definitions in § 2.1, we further formalize sparse/outcome and dense rewards in the context of LLM RL, according to the action and reward granularity, as shown in Table 2.

Token-Level Rewards. DPO [Rafailov et al., 2023] and its subsequent work [Rafailov et al., 2024] show that token-level rewards can be computed as log-likelihood ratios between the policy and the reference model. Implicit PRM [Yuan et al., 2025d] further shows that token-level rewards can be obtained by training an ORM and using the parameterization of Rafailov et al. [2024]. PRIME [Cui et al., 2025a] integrates ORM learning into RL training and uses implicit token-level rewards to train the policy. SRPO [Fei et al., 2025a] removes the ORM in PRIME and improves advantage estimation. Another line of works focus on using internal feedback as token-level rewards, such as token entropy [Cheng et al., 2025a, Tan and Pan, 2025] and strategic grams [Wang et al., 2025g].

Step-Level Rewards. Approaches to step-level rewards fall into two classes: *model-based* and *sampling-based*. Early works rely on human experts to annotate step-level dense rewards [Lightman et al., 2024, Uesato et al., 2022], which is costly and difficult to scale.

- **Model-based**: To reduce annotation cost, Math-Shepherd [Wang et al., 2024b] uses Monte Carlo estimation to obtain step-level labels and demonstrates that process verification with trained PRMs is effective in RL. PAV [Setlur et al., 2024] further improves process rewards via advantage modeling. To mitigate reward hacking with model-based step-level rewards, PURE [Cheng et al., 2025b] adopts min-form credit assignment rather than sum-form, while Tango [Zha et al., 2025] and AIRL-S [Jin et al., 2025c] jointly train the policy and PRMs. With the strong verification capabilities of generative PRMs [Zhao et al., 2025b] (discussed in § 3.1.2), ReasonFlux-PRM [Zou et al., 2025], TP-GRPO [He et al., 2025f], and CAPO [Xie et al., 2025b] leverage them to provide step-level rewards for RL training. Nevertheless, model-based dense rewards are vulnerable to reward hacking, and training PRMs online is expensive.

- **Sampling-based**: Another line of works use Monte Carlo sampling for online process reward estimation [Guo et al., 2025c, Hou et al., 2025, Kazemnejad et al., 2025, Li et al., 2025q, Yang et al., 2025f, Zheng et al., 2025c]. VinePPO [Kazemnejad et al., 2025] improves PPO via Monte Carlo estimation. To improve step segmentation, SPO [Guo et al., 2025c], TreeRL [Hou et al., 2025], and FR3E [Zheng et al., 2025c] use low-probability or high-entropy tokens as division points. To improve sample efficiency and advantage estimation, SPO [Guo et al., 2025c], TreeRPO [Yang et al., 2025f], TreeRL [Hou et al., 2025] and TreePO [Li et al., 2025q] explore tree-based structures for fine-grained process reward computation. MRT [Qu et al., 2025b], S-GRPO [Dai et al., 2025a], VSRM [Yue et al., 2025a], and SSPO [Xu et al., 2025f] force the LLM to terminate the thinking process at intermediate positions to estimate step-level rewards

efficiently. PROF [Ye et al., 2025a] utilizes the consistency between outcome rewards and process rewards to filter noisy data for RL training.

Turn-Level Rewards. Turn-level rewards evaluate each complete agent-environment interaction, such as a tool call and its result, providing feedback at the granularity of a single turn in multi-turn tasks. Research on turn-level rewards can be broadly divided into two lines: direct per-turn supervision and deriving turn-level signals from outcome-level rewards.

- For direct per-turn supervision, works provide explicit feedback at each turn. For example, Emotion-sensitive dialogue policy learning [Zhu et al., 2024] exploits user emotions as per-turn rewards to guide policy optimization, showing how turn-level feedback can enhance interaction quality in conversational agents. Similarly, ToolRL [Qian et al., 2025] designs structured rewards on format and correctness that are provided at each tool invocation step, offering dense turn-level signals for learning. Zeng et al. [2025d] further leverage verifiable signals with explicit turn-level advantage estimation to improve multi-turn tool use during RL. In addition, SWEET-RL [Zhou et al., 2025g] learns a step/turn-level critic that provides per-turn rewards and credit assignment, thereby supplying explicit turn-level supervision. More recently, MUA-RL [Zhao et al., 2025d] incorporates simulated user interactions into the RL loop, where each multi-turn exchange produces per-turn feedback, allowing the agent to iteratively refine its policy under realistic user-agent dynamics. G-RA [Sun et al., 2025g] extends this line of work by introducing gated reward aggregation, where dense turn-level rewards (e.g., action format, tool call validity, tool choice) are only accumulated if higher-priority outcome-level conditions are satisfied.

- For deriving turn-level signals from outcome-level rewards, the idea is to decompose or redistribute outcome-based supervision into finer-grained units. Aligning Dialogue Agents with Global Feedback [Lee et al., 2025] transforms session-level scores into turn-level pseudo-rewards, and GELI [Lee et al., 2024a] exploits multimodal cues such as prosody and facial expressions to refine session-level feedback into local turn-level signals. Similarly, SPA-RL [Wang et al., 2025e] redistributes outcome-based rewards into per-step or per-turn contributions through progress attribution. ARPO [Dong et al., 2025b] follows this line by attributing step/turn-level advantages from trajectory-level outcomes (e.g., after tool use), effectively converting global returns into localized signals.

Overall, turn-level rewards, whether directly assigned at each interaction or derived from outcome decomposition, serve as a bridge between process- and outcome-based supervision, and play a central role in stabilizing and improving optimization in multi-turn agent RL, with more details in § 6.2.

3.1.4. Unsupervised Rewards

Takeaways

- Unsupervised rewards eliminate the human annotation bottleneck, enabling reward signal generation at the scale of computation and data, not human labor.

- Main approaches include deriving signals either from the model's own processes (Model-Specific: consistency, internal confidence, self-generated knowledge) or from automated external sources (model-agnostic: heuristics, data corpora).

Frontier language models excel at a wide range of tasks, including many that are exceptionally challenging [Glazer et al., 2024, Jimenez et al., 2023, Li et al., 2024b, Phan et al., 2025]. However, a

key limitation in advancing these models is the reliance on human-generated reward signals for RL (§ 3.1.1–3.1.3). For tasks requiring superhuman expertise, human feedback is often slow, expensive, and impractical [Burns et al., 2023]. To address this, a promising approach is Unsupervised RL, which uses automatically generated, verifiable reward signals instead of ground-truth labels. This method is fundamental to achieving scalable RL for LLMs. This section surveys these unsupervised reward mechanisms, categorizing them into two types based on their source: those derived from the model itself (*Model-Specific*) and those from external, non-human sources (*Model-Agnostic*).

Model-Specific Rewards. This paradigm uses an LLM's internal knowledge as the sole source of supervision. It operates on the assumption that a high-performing model will generate consistent, confident, or evaluatively sound outputs. This method is highly scalable, requiring only the model and computational resources to generate a virtually infinite amount of "labeled" data. However, its closed-loop nature risks reward hacking and model collapse.

- **Rewards from Output Consistency**: This approach posits that correct answers will form a dense, consistent cluster among multiple generated outputs. Foundational works like EMPO [Zhang et al., 2025h] and Test-Time Reinforcement Learning (TTRL) [Zuo et al., 2025b] operationalize this via clustering and majority voting, respectively. Subsequent methods aim to refine this by improving efficiency (ETTRL [Liu et al., 2025c]), incorporating reasoning trajectories (CoVo [Zhang et al., 2025g]), or using contrastive agreement to combat reward hacking (Co-Reward [Zhang et al., 2025v]).

- **Rewards from Internal Confidence**: An alternative is to derive rewards directly from the model's internal states, using confidence as a proxy for correctness. Signals can be based on cross-attention (CAGSR [Kiruluta et al., 2025]), negative entropy (EM-RL [Agarwal et al., 2025b], RENT [Prabhudesai et al., 2025]), or generation probabilities (Intuitor [Zhao et al., 2025e], RLSC [Li et al., 2025g], RLSF [van Niekerk et al., 2025]). The success of these methods often depends on the base model's initial quality [Gandhi et al., 2025] and can be brittle [Press et al., 2024, Shumailov et al., 2023], as they rely on priors like low-density separation between correct and incorrect paths [Chapelle and Zien, 2005, Lee et al., 2013].

- **Rewards from Self-Generated Knowledge**: This paradigm uses the model's knowledge to create learning signals, either by acting as a judge (self-rewarding) or a problem proposer (self-instruction). In self-rewarding, the model evaluates its own outputs to generate a reward, a concept framed by Yuan et al. [2024] and Wu et al. [2024] and applied in works like SSR-Zero [Yang et al., 2025e] and MINIMO [Poesia et al., 2024]. In self-instruction, a proposer model generates a curriculum for a solver. The proposer is often rewarded for creating tasks of optimal difficulty [Chen et al., 2025i, Huang et al., 2025a, Zhao et al., 2025a], while the solver's reward can be model-agnostic (e.g., from a code executor in AZR [Zhao et al., 2025a]) or Model-Specific (e.g., via majority voting in SQLM [Chen et al., 2025i] and SeRL [Fang et al., 2025a]).

Model-Agnostic Rewards. In contrast to Model-Specific methods, this paradigm derives rewards from external, automated sources. This approach grounds the learning process in external information, eliminating the need for human labels. Its core principle is that these external signals are readily accessible and do not require manual effort. However, since precise feedback is often unavailable, the quality of the proxy reward is critical, and the risk of reward hacking persists.

- **Heuristic Rewards**: This approach constitutes another form of rule-based reward, employing simple, predefined rules based on output properties such as length or format as proxies for quality. It represents a specific case discussed in § 3.1.1. This was pioneered by DeepSeek-R1 [Guo et al.,

2025a] and later refined with techniques like dynamic reward scaling [Yu et al., 2025d]. While scalable, these heuristics can be gamed by the model, leading to superficial improvements without advancing true capability [Liu et al., 2025s, Xin et al., 2025].

- **Data-Centric Rewards**: This approach derives reward signals from the structure of large, unlabeled corpora. Analogous to next-word prediction for large-scale pre-training, RPT [Dong et al., 2025c] reframes next-token prediction as an RL task, turning web-scale datasets into millions of training examples. At a meta-level, SEAL [Zweiger et al., 2025] allows a model to generate its own training data and hyperparameters, using downstream performance as the reward.

In summary, unsupervised reward design is essential for creating scalable RL systems for LLMs. The Model-Specific paradigm facilitates self-improvement by leveraging the model's internal knowledge, whereas the Model-Agnostic paradigm grounds learning in external, automated feedback. While both approaches effectively bypass the human annotation bottleneck, they remain susceptible to reward hacking [Zhang et al., 2025p]. The future of scalable RL will likely involve hybrid systems that strategically combine these methods, for instance, using data-centric rewards for pre-training, Model-Specific self-rewarding for fine-tuning on complex reasoning, and minimal human oversight for safety and alignment.

3.1.5. Rewards Shaping

> **Takeaways**
>
> - Reward shaping enriches sparse signals into stable, informative gradients for LLM training.
>
> - Combine verifiers with reward models, and use group baselines plus *Pass@K*-aligned objectives to stabilize training, expand exploration, and match evaluation metrics at scale.

As noted, the primary learning objective of agents in RL is to maximize cumulative rewards, making the design of the reward function particularly critical [Sutton et al., 1998]. In previous sections, we introduced various reward functions, such as verifiable rewards (§ 3.1.1), generative rewards (§ 3.1.2), dense rewards (§ 3.1.3) and even unsupervised rewards (§ 3.1.4). Beyond reward engineering, it is equally important to consider how the reward function can be modified or augmented to encourage behaviors that drive progress toward the desired solution. This process, known as reward shaping [Goyal et al., 2019, Gupta et al., 2022, Hu et al., 2020, Xie et al., 2023], can be categorized into rule-based and structured-based reward shaping.

Rule-based Reward Shaping. The simplest and most commonly adopted approach to reward shaping in LLM-based RL involves combining rewards from both a rule-based verifier and a reward model to generate the overall reward signal, as demonstrated in Qwen2.5 Math [Yang et al., 2024a]. Typically, a constant coefficient is used to balance the contributions of the reward model and the rule-based component. Rather than assigning identical rewards to all correct responses, this method allows for further ranking of responses based on the scores from the reward model. This approach is particularly useful for more challenging samples and helps to avoid cases where all reward values are 0 or 1, which would otherwise lead to ineffective learning gradients [Yu et al., 2025d]. This heuristic combination strategy is widely employed in open-domain tasks, where integrating rule-based rewards and reward models [Guo et al., 2025b, Liao et al., 2025a, Liu et al., 2025w] results in more informative and effective reward signals for the RL of LLM [Su et al., 2025c, Zeng et al., 2025c, Zhang et al., 2024a]. Another approach involves combining rule-based rewards, such as outcome-level rewards and format rewards, as implemented in DeepSeek-R1 [Guo et al., 2025a], which enables LLMs to learn long chain-of-thought reasoning. These rewards include format-based [Xin et al., 2025] and length-based

components [Liu et al., 2025o] to address various exceptions in the outputs of LLMs. Recent work also explores multi-role RL training and assigns different rewards for different roles with different reward functions, such as solver and critic [Li et al., 2025h]. Typically, these rewards are combined using manually set constants. Recent works have also explored multi-role RL training [Li et al., 2025h,i], assigning distinct reward functions to different roles to encourage diverse behaviors and objectives [Li et al., 2025h], such as solver and critic.

Structure-based Reward Shaping. In contrast to rule-based reward shaping, which relies solely on individual samples, structure-based reward shaping computes rewards across a group of candidates by leveraging list-wise or set-level baselines. One influential method is GRPO [Shao et al., 2024], which uses the group mean of responses to the same question G as a baseline (or variants such as leave-one-out [Ahmadian et al., 2024] or ranking) and constructs advantages accordingly for PPO-style updates [Schulman et al., 2017b]. Recent works have further modified the optimization objective or credit allocation strategies to promote stronger exploration and achieve closer alignment with evaluation metrics, such as *Pass@K* [Yue et al., 2025b]. For example, Walder and Karkhanis [2025] perform a joint transformation on the final reward, making the optimization directly equivalent to set-level objectives like *Pass@K*, and provide low-variance, unbiased gradient estimation. Chen et al. [2025v] directly target *Pass@K* in deriving and analyzing advantages and efficient approximations, decomposing set-level targets back into individual sample credit allocation. Reward shaping methods in this direction aim to stabilize training and encourage the policy to explore more extensively, thereby reducing the risk of premature convergence to suboptimal local solutions.

3.2. Policy Optimization

In this subsection, we first provide a technical overview of the mathematical formulation of the policy gradient objective (§ 3.2.1). Next, we divide the on-policy optimization algorithms in RL into two categories based on how the reward is generated for the gradient calculation process: critic-based (§ 3.2.2) and critic-free (§ 3.2.3). In addition, we discuss recent studies that combine on-policy RL with offline datasets for more sophisticated post-training (i.e., off-policy) optimization (§ 3.2.4), as well as various regularization techniques such as entropy and KL (§ 3.2.5).

3.2.1. Policy Gradient Objective

As introduced in § 2.1, the context in RL for LLMs is treated as the environment, and the probability distribution of the next-level prediction is treated as a policy. For an RL system, the objective of the system is to find an optimal policy such that the expected cumulative reward generated by the system is maximized. The RL policy optimization algorithms for LLMs are mostly first-order gradient-based algorithms, due to the large number of parameters in the LLMs. In general, RL algorithms seek to optimize network parameters such that the expected reward is maximized. Below, we present a general formulation for LLM gradient calculation of RL algorithms.

Notations. Although we have introduced the relevant symbols in § 2.1, we revisit these definitions here for the sake of comparative clarity. Let $x \sim \mathcal{D}$ be a prompt (initial state $s_1 = s$). A stochastic policy π_θ generates a sequence $y = (a_1, \cdots, a_T)$, we denote the total sequence length of y as $|y|$, with states defined by $s_{t+1} = (x, s_{\leq t})$. We assume a primarily sequence-level reward $R(x, y)$, optionally decomposed into token-level rewards r_t. We collect $G \geq 1$ responses per prompt using a *behavior policy* π_b (also denoted as π_{old}, referring to an earlier version of the current policy). Optionally, a *reference policy* π_{ref} (e.g., base, finetuned or instructed models) may be used for regularization.

We revisit the MDP defined in § 2.1. In MDPs, we denote the expected cumulative reward given

the current state s as the V (value) function

$$V(s) = \mathbb{E}_{a_t \sim \pi_\theta(s_t), s_{t+1} \sim \mathcal{P}(s,a)} \Big[\sum_{t=0}^{T} \gamma^t r(s_t, a_t)|s_0 = s\Big], \tag{2}$$

and the expected cumulative reward for the current state-action pair is denoted as Q (quality) function

$$Q(s,a) = \mathbb{E}_{a_t \sim \pi_\theta(s_t), s_{t+1} \sim \mathcal{P}(s,a)} \Big[\sum_{t=0}^{T} \gamma^t r(s_t, a_t)|s_0 = s, a_0 = a\Big]. \tag{3}$$

Then the objective of RL can be formulated as a maximization problem for the expected cumulative reward. To optimize the objective function, it is a common practice to use the Policy Gradient algorithm [Sutton et al., 1999, Williams, 1992] for gradient estimation:

$$\nabla_\theta \mathcal{J}(\theta) = \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta} \left[\sum_{t=1}^{T} \nabla_\theta \pi_\theta(y_t|y_{<t}) Q_t\right]. \tag{4}$$

The policy gradient can be justified by the intuition that an algorithm following the policy gradient should maximize the probability of better-than-average actions and minimize the probability of worse-than-average actions. This notion led to the introduction of the A (advantage) function $A(s,a) = Q(s,a) - V(s)$. The advantage measures how much the current action improves upon the expected total reward compared to the existing policy. The advantage can be estimated in many ways. If we only have rewards for the full trajectory, the vanilla REINFORCE algorithm [Williams, 1992] directly defines $A_t = R(x, y)$.

For the case of training LLMs, the vanilla policy gradient algorithms often suffer from stability issues. Instead, the training is often done with the PPO algorithm [Schulman et al., 2017b]. For an algorithm with N samples, we define a general objective with PPO-style updates as follows:

$$\mathcal{J}(\theta) = \mathbb{E}_{\text{data}} \left[\frac{1}{Z} \sum_{i=1}^{N} \sum_{t=1}^{T_i} \min\left(w_{i,t}(\theta)\hat{A}_{i,t}, \text{clip}(w_{i,t}(\theta), 1 - \epsilon_{\text{low}}, 1 + \epsilon_{\text{high}})\hat{A}_{i,t}\right)\right], \tag{5}$$

where:

- $w_{i,t}(\theta)$ is the importance ratio;

- $\hat{A}_{i,t}$ is the advantage (either token-wise or sequence-level);

- T_i is the number of tokens or responses per sample;

- N is the total number of samples under the given prompt;

- Z is the normalization factor (e.g., total tokens, group size, etc.).

The PPO algorithm [Schulman et al., 2017b] was first proposed as a computationally efficient approximation for the TRPO algorithm [Schulman et al., 2015a]. PPO excels when vanilla policy gradient methods suffer from poor data efficiency and robustness issues. In addition, PPO is shown to be much simpler to implement, more general, and has better sample complexity compared to TRPO.

However, since the complex and long CoT nature of LLMs, the exact objective function, gradient estimation, and update techniques can take a wide range of different forms as shown in Table 3.

3.2.2. Critic-based Algorithms

Table 3 | Comparison of representative RL algorithms for reasoning models training.

Date	Algorithm	Advantage Estimate	Importance Sampling	Loss Agg.
2017.01	PPO	Critic-GAE	PPO-Style	Token-Level
2023.10	ReMax	Greedy Baseline	N/A	Token-Level
2024.02	RLOO	Leave-One-Out	N/A	Token-Level
2025.01	RF++	Negative KL + Batch Relative	PPO-Style	Sequence-level
2024.02	GRPO	Group Relative	PPO-Style	Sequence-level
2025.01	PRIME	Outcome + Implicit PRM	PPO-Style	Token-Level
2025.03	VAPO	Value Adjusted GAE	Clip-Higher	Token-Level
2025.03	Dr. GRPO	Group Baseline	PPO-Style	Token-Level
2025.04	DAPO	Group Relative	Clip-Higher	Token-Level
2025.05	Clip-Cov	Group Relative	PPO-Style	Sequence-level
2025.05	KL-Cov	Group Relative	PPO-Style	Sequence-level
2025.06	CISPO	Group Relative	Clipped IS-weight	Token-Level
2025.07	GSPO	Group Relative	PPO-Style	Sequence-level
2025.08	GMPO	Group Relative	Clip-Wider	Geometric-Avg
2025.08	GFPO	Filter + Group Relative	PPO-Style	Token-level
2025.08	LitePPO	Group-level mean, Batch-level std	PPO-Style	Token-level
2025.08	FlashRL	Group Relative	Truncated IS	Token-level

Takeaways

- The critic model is trained on a small subset of labeled data, and provides scalable token-level value signals for unlabeled roll-out data.

- The critic is required to run and update alongside the LLM, resulting in a significant computational overhead and scales unfavorably for complex tasks.

The first LLM-related works in RL focus on how to effectively align the LLM policy to the external supervision, to make LLMs have better instruction following capabilities while ensuring the models are helpful, honest, and harmless. The most common approach for LLM alignment is RLHF [Bai et al., 2022a, Christiano et al., 2017, Ouyang et al., 2022, Stiennon et al., 2020]. This technique utilizes humans as a critic for the learning algorithm; the exact steps are as follows. First, a selection of model outputs is generated by the LLM and labeled by humans to create a dataset. The dataset is then used to train a reward model to predict which response would be preferred by humans. Lastly, the reward model is used to train the LLM along with a value function, acting as the critic in the system. The training is often done with the PPO algorithm [Schulman et al., 2017b]. The PPO algorithm formulates the objective in the following form:

$$\mathcal{J}_{\text{PPO}}(\theta) = \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_{\theta_{old}}(\cdot|x)} \left[\frac{1}{|y|} \sum_{t=1}^{|y|} \min \left(w_t(\theta)\hat{A}_t, \, \text{clip}(w_t(\theta), 1 - \epsilon, 1 + \epsilon)\hat{A}_t \right) \right], \quad (6)$$

where $\hat{A}_t$ is a value-model-based advantage and

$$w_t(\theta) = \frac{\pi_\theta(y_t|x, y_{<t})}{\pi_{\theta_{old}}(y_t|x, y_{<t})}. \quad (7)$$

We note that PPO is proposed as a clipped surrogate objective of TRPO, which preserves the conservative policy iteration of TRPO while being unconstrained and having a computational complexity

close to traditional policy gradient methods. Due to the discrepancy between the current policy and the sampling distribution, the advantage in TRPO is multiplied by w_t, the importance sampling factor in Equation 6. PPO maximizes the same objective as TRPO, but removes the trust region constraint. Furthermore, PPO adds a clipping mechanism and a KL regularization factor to ensure the current policy does not diverge too far from the rollout policy $pi_{\theta_{old}}$.

In critic-based approaches, the scalability of RL is achieved by the introduction of a critic model. After the reward model is sufficiently trained on the manually labeled small subset of generated data, it can be used to construct the critic model, generating token-level value signals on a much larger scale for the vast majority of unlabeled generated data for RL. However, these works require a critic model to run and optimize along the target LLM, and create a significant computational overhead.

In PPO, the critic model adapts the Generalized Advantage Estimator (GAE) [Schulman et al., 2015b] from the RL literature. GAE is typically constructed with the temporal difference error

$$\delta_t = r_t + \gamma V(y_{t+1}) - V(y_t), \tag{8}$$

which is then accumulated across time steps:

$$\hat{A}_{GAE,t} = \sum_{l=t}^{T} (\gamma\lambda)^l \delta_{t+l}, \tag{9}$$

where γ is the discount factor of the MDP and λ is a parameter that controls the bias-variance tradeoff.

Recent work has argued that the decay factor scales unfavorably for complex reasoning tasks that require long CoT and proposed a Value-Calibrated PPO [Yuan et al., 2025f] and VAPO [Yue et al., 2025c], VRPO [Zhu et al., 2025a] proposed novel mechanisms for enhancing the robustness of the critic model under noisy reward signals.

In addition, critic-based algorithms [Hu et al., 2025b] have also demonstrated steady scalability properties for Monte-Carlo estimation with rule-based rewards. Similar approaches have been adapted with fixed external models [Lu et al., 2024, Wang et al., 2024b] by the implementation of PRMs.

Another approach to introduce critic models is done with the introduction of Implicit PRM [Yuan et al., 2025d]. This approach is also able to provide token-level supervision for scalable RL training. Different from the GAE approach, methods such as Implicit PRM [Yuan et al., 2025d] and PRIME [Cui et al., 2025a] adapted a specific reward model formulation to directly generate token-level rewards.

3.2.3. Critic-Free Algorithms

> **Takeaways**
>
> - Critic-free algorithms only require sequence-level rewards for training, making them more sufficient and scalable.
>
> - For RLVR tasks, rule-based training signals reliably prevent critic-related issues such as reward hacking.

Apart from the critic-based models, which provide token-level feedback signals for model training, many recent works have stated that the response-level rewards are sufficient for scalable reasoning tasks with RL. These critic-free algorithms apply the same rule-based or model-generated response-level reward for all tokens in the response and demonstrate their effectiveness across various tasks. Compared to the critic-based algorithms, critic-free approaches do not require a separate critic model,

significantly reducing the computational requirement and simplifying training. Moreover, when training LLMs in rule-based environments where the reward for any response can be clearly defined, critic-free algorithms can avoid reward hacking issues that may arise from an ill-trained critic model. This property makes critic-free algorithms more scalable than critic-based approaches in such settings.

The classic REINFORCE [Williams, 1992] algorithm was among the first algorithms developed for RL. It was applied to the LLM problem in [Ahmadian et al., 2024]. The exact formulation for REINFORCE is as follows,

$$\mathcal{J}_{\text{REINFORCE}}(\theta) = \mathbb{E}_{x \sim \mathcal{D}, \{y\} \sim \pi_{old}(\cdot|x)} \left[R(x, y) \nabla_\theta \log(\pi_\theta(y|x)) \right], \tag{10}$$

where $R(x, y)$ usually takes the form of ± 1 for RLVR tasks. This naive formulation takes the entire sequence as a single action and considers the response task as a bandit. However, the vanilla algorithm usually suffers from severe instability issues due to high variance. ReMax [Li et al., 2023c] introduced a variance reduction mechanism to REINFORCE with a greedy baseline estimation. Ahmadian et al. [2024] also introduced RLOO, which further provides an unbiased baseline with more stable results. REINFORCE++ [Hu, 2025] adapts techniques such as clipping and global advantage normalization from PPO and GRPO style algorithms to provide a more accurate advantage and gradient estimations.

One of the most popular critic-free approaches for RL is GRPO [Shao et al., 2024]. The objective formulation for GRPO is as follows,

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{x \sim \mathcal{D}, \{y_i\}_{i=1}^G \sim \pi_{\theta_{old}}(\cdot|x)} \left[\frac{1}{G} \sum_{i=1}^G \frac{1}{|y_i|} \sum_{t=1}^{|y_i|} \min \left(w_{i,t}(\theta)\hat{A}_{i,t}, \text{clip}(w_{i,t}(\theta), 1 - \epsilon, 1 + \epsilon)\hat{A}_{i,t} \right) \right], \tag{11}$$

$$w_{i,t}(\theta) = \frac{\pi_\theta(y_{i,t}|x, y_{i,<t})}{\pi_{\theta_{old}}(y_{i,t}|x, y_{i,<t})}, \quad \hat{A}_{i,t} = \hat{A}_i = \frac{R(x, y_i) - \text{mean}(\{R(x, y_i)\}_{i=1}^G)}{\text{std}(\{R(x, y_i)\}_{i=1}^G)}, \tag{12}$$

where all the tokens in y_i share the same advantage as $\hat{A}_i$.

GRPO is a critic-free modification of PPO, where instead of GAE provided by a critic, the entire sequence uses the same advantage estimate, which is calculated by a group-relative normalization as a better estimation than the binary rule-based reward. Compared to PPO and REINFORCE-style methods, the group-based advantage calculation of GRPO effectively reduces variance from training signals and has been shown to speed up the training process. Other recent approaches, including DAPO [Yu et al., 2025d], CISPO [Chen et al., 2025a], Dr. GRPO [Liu et al., 2025t], LitePPO [Liu et al., 2025v], made further modifications to GRPO with careful tuning of sampling strategy, clipping threshold, and loss normalization to further enhance the stability of the RL training process. Another recent approach, GSPO [Zheng et al., 2025a], replaces the token-wise clipped importance sampling ratio with a sequence-level clipping.

Apart from REINFORCE and GRPO-related algorithms, there are other critic-free approaches. VinePPO modifies PPO by replacing the learned critic with a Monte Carlo advantage estimation. CPGD [Liu et al., 2025y] proposed a novel policy gradient objective, along with a drift regularization mechanism. K1.5 [Team, 2025d] utilizes RL with an adaptation of mirror descent in the training of foundational models, which successfully enhanced the long-context reasoning capabilities of LLMs. Lv et al. [2025] have recently introduced a unified policy gradient estimator with a hybrid post-training algorithm, providing a unified framework for policy gradient estimation for RL in LLMs.

Importance Sampling for Policy Optimization. Due to the rollout-reward-training cycle for RL, it is generally computationally intractable to ensure the rollout data follows the exact policy distribution of the current model. Therefore, importance sampling was introduced to reduce bias in training. The first version of importance sampling in RL was introduced in TRPO, where a token-wise importance

ratio $w_{i,t}$ was introduced into the objective. This approach is widely adopted among recent works, such as GRPO. This approach is restricted to the token-wise importance ratio since the actual distribution ratio can not be effectively calculated over the long context of CoT. However, token-level importance sampling introduces another bias into RL algorithms, since the actual sampling distribution given policy is defined with respect to the state-action pair, whereas the token-level approach only considers the current action. GMPO [Zhao et al., 2025f] seeks mitigation by introducing a geometric averaging to increase training robustness for tokens with extreme importance sampling ratios. In the recent work of GSPO [Zheng et al., 2025a], a sequence-level importance sampling factor was calculated. GSPO adds a unique normalization factor to ensure that the probability ratio can be calculated, but this approach is also a biased estimation of the actual importance sampling factor. A promising new direction is to move beyond the theoretical framework of standard on-policy policy gradient methods and instead derive inherently off-policy algorithms directly from supervised learning theory [Chen et al., 2025c]. We will provide a detailed introduction to off-policy optimization in the next section.

3.2.4. Off-policy Optimization

> **Takeaways**
>
> - Off-policy RL boosts sample efficiency by decoupling data collection from policy learning, enabling training from historical, asynchronous, or offline datasets.
>
> - Modern practice mixes off-policy, offline, and on-policy methods (e.g., SFT+RL or large-scale offline learning) to improve stability and performance.

In RL, off-policy methods address the scenario where the policy being learned (the target policy) differs from the policy generating the data (the behavior policy). This core distinction allows an agent to learn about an optimal course of action without having to follow it during data collection. This flexibility is a key advantage, often leading to more sample-efficient algorithms than on-policy counterparts, which require new data sampled directly from the current policy for each update. A core challenge in these methods is correcting for the distributional shift between the behavior policy and the target policy, often addressed using importance sampling with a weighted objective function:

$$\mathcal{L}_{\text{policy}}(\theta) = -\mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_b(y|x)} \left[\frac{\pi_\theta(y|x)}{\pi_b(y|x)} \cdot r(x, y) \right], \tag{13}$$

where the fraction $\frac{\pi_\theta(y|x)}{\pi_b(y|x)}$ serves as the importance weight between the target policy π_θ and the behavior policy π_b.

In practical large-scale model training, off-policy learning often manifests in different forms. Recent works can be broadly grouped into three aspects: 1) training–inference precision discrepancies, where models are trained with high precision but deployed in lower precision, creating a gap between the target and behavior policies; 2) asynchronous experience replay mechanisms, which enhance efficiency and stability by reusing past trajectories during learning; and 3) broader off-policy optimization approaches, including optimizer-level improvements, data-level offline learning, and hybrid methods that combine supervised fine-tuning with RL.

Training-Inference Precision Discrepancy. A notable off-policy scenario arises from the difference in parameter precision between the training model and the inference model, a common consequence of using different frameworks for training and inference [Yao et al., 2025a] (e.g., vLLM vs. FSDP), or of model quantization to accelerate inference [Lin et al., 2016]. It is common practice to train a model using high-precision parameters (e.g., 32-bit floating point) and then deploy a quantized version with lower-precision parameters (e.g., 8-bit integers) [Liu et al., 2025h]. This creates a discrepancy where

the deployed, low-precision model acts as the behavior policy, generating real-world interaction data, while the high-precision model remains the target policy being updated during training. While this mismatch establishes an off-policy learning problem, research indicates that the policy divergence due to quantization is often minimal. Consequently, this difference can be effectively managed with simple correction techniques, such as truncated importance sampling (TIS) [Ionides, 2008, Yao et al., 2025a], allowing for stable training while retaining the benefits of accelerated inference.

Asynchronous Off-Policy Training. Asynchronous training pairs naturally with off-policy RL for LLMs. Many actors generate trajectories concurrently and append them to a shared replay buffer, while a centralized learner samples mini-batches from this buffer to update the target policy. Building on this view, several recent methods deliberately reuse past trajectories to improve efficiency and stability. One example is Retrospective Replay [Dou et al., 2025], which enhances exploration for LLM reasoning by selectively replaying earlier reasoning traces to guide current policy updates. Similarly, EFRame [Wang et al., 2025b] adopts an exploration-filter-replay mechanism, interleaving filtered responses with fresh rollouts to encourage deeper reasoning. In the domain of code generation, Possibility- and Pass-rate Prioritized Experience Replay (P2Value) [Chen et al., 2024c] takes this further by prioritizing high-value code samples in the buffer, leading to more stable optimization. Extending these ideas to multimodal interaction, ARPO [Lu et al., 2025b] applies replay to GUI agents, where successful trajectories are reused to provide reliable learning signals under sparse rewards. Finally, RLEP [Zhang et al., 2025c] anchors exploration with an experience buffer of verified successful trajectories from earlier runs, which are blended with new rollouts to balance reliability with discovery. Together, these approaches illustrate how replay buffers have become a cornerstone of modern, asynchronous off-policy training for LLM-based agents.

Off-Policy Optimization. Recent advancements in fine-tuning LLMs have explored sophisticated optimization strategies beyond traditional on-policy RL. These methods, broadly categorized as off-policy and mixed-policy optimization, aim to improve sample efficiency, training stability, and overall performance by creatively using data from various sources. We introduce this topic below:

- **Optimizer-Level Off-Policy Methods:** These approaches focus on improving the optimization procedure itself, emphasizing stability and efficiency in policy updates. For example, SPO [Cohen et al., 2025] introduces a soft policy optimization method that enables stable online, off-policy RL, while TOPR [Roux et al., 2025] proposes a tapered off-policy REINFORCE algorithm for improved stability and efficiency. ReMix [Liang et al., 2025a] further highlights this by focusing on efficiently leveraging off-policy data to maximize the utility of available information.

- **Data-Level Off-Policy Methods:** A class of off-policy algorithms learns entirely from large-scale, external offline data [Zhang et al., 2025f]. For instance, the Decision Field Theory (DFT) framework [Wu et al., 2025i] adapts methodologies from other fields to learn from these complex datasets. Similarly, techniques like Implicit Fine-Tuning (IFT) [Hua et al., 2024] are being explored to refine pre-trained models using this external data, aiming to enhance their performance on specific downstream tasks. Another relevant method is DPO [Rafailov et al., 2023], which directly optimizes a policy from preference data through a simple classification objective. These methodologies collectively represent a move towards more data-centric approaches in RL, enabling the development of sophisticated policies from vast and diverse sources of offline data.

- **Mix-Policy Methods:** In parallel with reusing past data more efficiently, mixed-policy optimization represents another significant trend, which combines the strengths of SFT and RL. This hybrid approach leverages the stability from SFT on expert data while using RL to optimize for specific reward functions, integrating the supervised data in two primary ways. One strategy is at the loss-level, where SFT and RL objectives are combined directly in the loss function [Lv et al., 2025,

Xiao et al., 2025b, Zhang et al., 2025j]. Methods like UFT [Liu et al., 2025j], SRFT [Fu et al., 2025c], LUFFY [Yan et al., 2025a], RED [Guan et al., 2025], and ReLIFT [Ma et al., 2025a] all exemplify this by creating unified or single-stage training processes that learn from both expert demonstrations and RL feedback simultaneously. A second strategy operates at the data level, using expert data to structure the generation process itself. Here, high-quality data serves as a prefix or anchor to guide the model's exploration [Guo et al., 2025d]. For instance, BREAD [Zhang et al., 2025o] generates branched rollouts from expert anchors, and Prefix-RFT [Huang et al., 2025g] blends the training regimes via prefix sampling. By mixing policies at either the loss or data level, these methods prevent reward hacking and ensure the model retains knowledge from SFT, leading to more robust and capable models for complex reasoning.

3.2.5. Regularization Objectives

> **Takeaways**
>
> - Objective-specific regularization helps balance exploration and exploitation, boosting RL efficiency and policy performance.
>
> - The optimal choice and form of KL, entropy, and length regularization remain open questions, each affecting policy optimization and scalability.

As introduced in previous sections, ensuring stability and preventing catastrophic policy drift is paramount. In particular, for long-horizon training, techniques such as KL regularization and entropy regularization are widely employed.

KL Regularization. The role of KL divergence regularization is a highly controversial topic in this area. In most studies, KL regularization is applied to 1). current policy π_θ and the reference policy π_{ref}, 2). current policy π_θ and the old policy π_{old}. We provide a unified formulation in Equation 14.

$$\mathcal{L}_{KL} = \beta \sum_{t=1}^{|y|} KL(\pi_\theta(\cdot|y_t)||\pi_{ref/old}(\cdot|y_t)). \tag{14}$$

- For the former, this is a commonly used technique in RLHF [Ouyang et al., 2022, Touvron et al., 2023]. It was initially introduced to prevent the model from being destructively updated. Prior work argues that incorporating a KL penalty is essential for maintaining stability and avoiding entropy collapse over thousands of training steps. To reduce the risk of the KL term excessively constraining progress, Liu et al. [2025i] use this method combined with a periodic reference policy reset, in which the reference model is updated to a recent snapshot of the training policy. To simultaneously maintain knowledge and enhance reasoning capabilities, Wang et al. [2025h] apply stronger KL regularization to low-entropy tokens and weaker regularization to high-entropy tokens. However, in the context of RL for reasoning with LLMs, which is more challenging than standard RLHF, the necessity of this kind of KL regularization needs to be reconsidered. Recently, many studies have identified that the policy is expected to explore freely during training, thus may diverge significantly from its initialization to discover new CoT structures, making the KL constraint an unnecessary restriction. Thus, a majority of other recent works advocate for removing the KL penalty entirely [An et al., 2025, Arora and Zanette, 2025, Chen et al., 2025q, Cui et al., 2025a, Fan et al., 2025b, He et al., 2025d, Liao et al., 2025b, Liu et al., 2025t, Yan et al., 2025a, Yu et al., 2025d] to simplify implementation, reduce memory cost and achieve more scalable GRPO.

- For the latter case, it can serve as a substitute for the clip form of the policy loss [Schulman

et al., 2017b]. Zhang et al. [2025q] discuss the differences between forward KL, reverse KL, normalized KL, and Normalized forms. This approach has also been adopted in Cui et al. [2025b], Lyu et al. [2025], Team [2025d], demonstrating its potential across different RL training scales. Nevertheless, its deeper mechanisms and its significance for scalable RL remain under exploration.

Entropy Regularization. In the RL literature, preserving policy entropy is widely considered a critical aspect of many algorithms [Eysenbach and Levine, 2021, Williams, 1992, Williams and Peng, 1991]. To this end, policy entropy is actively controlled through regularization techniques [Haarnoja et al., 2018, Schulman et al., 2017b, Ziebart et al., 2008].

$$\mathcal{L}_{\text{ent}} = -\alpha \sum_{t=1}^{|y|} H[\pi_\theta(\cdot|y_t)] = \alpha \sum_{t=1}^{|y|} \sum_{v=1}^{|\mathcal{V}|} \pi_\theta(y_t^v|y_t) \log \pi_\theta(y_t^v|y_t). \tag{15}$$

However, in RL for LLMs, directly applying entropy regularization is neither common nor effective [Cui et al., 2025b, He et al., 2025d]. The use of an explicit entropy regularization term in the loss function remains a point of contention. While some find it beneficial, using either a standard coefficient [Shrivastava et al., 2025] or a targeted loss function [Wu et al., 2025e], others argue against it, finding it can lead to instability or even training collapse, especially with sparse rewards [An et al., 2025, Liao et al., 2025b]. Many studies have shown the phenomenon of entropy collapse when no intervention is applied [Cheng et al., 2025a, Cui et al., 2025b, Yu et al., 2025d], which hinders effective policy exploration during training. To address it, He et al. [2025d] dynamically adjust the coefficient of the entropy loss, Yu et al. [2025d] employs the clip-higher technique to involve more low-probability tokens in the policy update, Wang et al. [2025l] directly train on 20% high-entropy tokens, Cheng et al. [2025a] and Chen et al. [2025j] emphasize entropy through incorporate it into the advantage computation. Beyond these techniques, which explicitly maximize entropy, Cui et al. [2025b] provide a theoretical explanation for the underlying mechanism of entropy dynamics, identifying the covariance between an action's output probability and its advantage as the entropy "driver". Built on this insight, Clip-Cov and KL-Cov are proposed to regulate entropy by selectively constraining a small portion of tokens exhibiting exceptionally high covariance.

Length Penalty. Recent successes of LRMs on complex tasks have validated the effectiveness of long-CoT reasoning. Yet longer reasoning traces incur higher inference costs. To balance the reasoning budget and performance [Agarwal et al., 2025a, He et al., 2025e], many works seek to reduce the reasoning cost while retaining the model performance [Aggarwal and Welleck, 2025, Liu et al., 2025o, Luo et al., 2025a, Su et al., 2025b, Xiang et al., 2025]. For example, Aggarwal and Welleck [2025] control reasoning length by ensuring adherence to user-specified length constraints, while Yuan et al. [2025a] and Luo et al. [2025a] design relative-length regularization and an accuracy-preservation constraint to the optimization objective, Xiang et al. [2025] and Liu et al. [2025o] propose to apply adaptive length penalties conditioned on problem difficulty to preserve the model ability.

3.3. Sampling Strategy

Unlike static datasets, RL depends on actively curated rollouts, where decisions about what and how to sample directly influence learning efficiency, stability, and the quality of acquired reasoning behaviors. Effective sampling strategies not only ensure diverse and informative training signals but also align the learning process with the intended reward structure and policy objectives. In this subsection, we survey recent advances in dynamic and structured sampling (§ 3.3.1), as well as hyperparameter adjustment techniques that further optimize sampling and policy improvement (§ 3.3.2).

3.3.1. Dynamic and Structured Sampling

> **Takeaways**
>
> - High-quality, diverse rollouts stabilize RL training and enhance overall performance by exposing agents to a broader range of meaningful experiences.
>
> - Balancing the exploration of diverse trajectories with maintaining high sampling efficiency presents a fundamental trade-off in RL.

Sampling has become a first-class lever in RL fine-tuning for reasoning LLMs, serving as an efficient and adaptive mechanism to maximize data utilization, reduce wasted computation, and enhance training effectiveness or a control and a guidance for LLMs to sample in a structured format.

Dynamic Sampling. Dynamic sampling adapts both the selection of prompts for rollout and the computational budget allocated to each, based on online learning signals such as success rate, advantage, uncertainty, or estimated difficulty. The primary goal is to concentrate computing on informative examples while avoiding saturated or unproductive ones. Existing methods generally fall into two categories:

- **Efficiency-oriented Sampling:** Some works use online-filtering to concentrate training on questions of medium difficulty to ensure training effectiveness and efficiency. A representative design is PRIME [Cui et al., 2025a], which applies an online filter to drop out too easy or too difficult problems. Another example is DAPO [Yu et al., 2025d], which over-samples and filters prompts whose rollouts are saturated (all-correct) or degenerate (all-wrong), then repeatedly samples until each mini-batch contains prompts with non-zero advantage, focusing on medium-difficulty cases to maintain informative gradients. Building on this foundation, prioritized schemes allocate rollout budget toward under-mastered items by sampling proportional to failure rates, as $p(i) \propto (1 - s_i)$ rule [Team, 2025d]. Curriculum learning approaches operate at multiple scales: category-level selection [Chen et al., 2025o] uses non-stationary bandits, while E2H [Parashar et al., 2025] follows easy-to-hard schedules with convergence guarantees for small models. Efficiency methods include pre-rollout selection to skip unhelpful prompts and difficulty-based online selection with rollout replay [Sun et al., 2025e, Zheng et al., 2025b]. POLARIS [An et al., 2025] formalizes this via offline difficulty estimation, constructing "mirror-J" distributions by model scale, continuously removing mastered items, and applying in-batch information replacement. Extending these efficiency gains, recent advances use lightweight controllers for adaptive sampling [Do et al., 2025, Shi et al., 2025b] without modifying algorithms, while experience replay with random reshuffling [Fujita, 2025] reduces variance through balanced utilization, and enhanced prioritized methods [Li et al., 2024a] dynamically adjust priority weights based on experience pool features. Sampling efficiency can also be improved by structuring the generation process with expert data: high-quality demonstrations are used as prefix anchors to bias exploration toward promising regions of the search space [Guo et al., 2025d, Huang et al., 2025g, Zhang et al., 2025o]. The field shifts from uniform sampling to model-aware strategies combining item-, category-, and difficulty-level choices for stronger learning signals per rollout.

- **Exploration-oriented Sampling:** There are other works aiming for exploration using dynamic rollout. ARPO [Dong et al., 2025b] is proposed to implement entropy-guided rollout to ensure high uncertainty so that the model will call external tools, improving diversity. DARS [Yang et al., 2025g] proposes a rollout mechanism to dynamically assign sample numbers for questions of different difficulty. Zhou et al. [2025f] propose RuscaRL by providing the policy with different rubrics during rollout to enhance exploration. Different from above, G^{2}RPO-A [Guo et al., 2025d]

does not drop all-wrong questions, but add a guidance during the thinking process to generate correct samples for hard questions. Besides, Li et al. [2025s] utilize the latest k checkpoints to generate k responses to prevent forgetting during training.

Structured Sampling. Structured sampling controls not only what is sampled but also the topology of reasoning traces, aligning generation, credit assignment, and compute reuse with the underlying structure of problem solving. By organizing rollouts as trees or through shared and segmented prefixes, these methods enable node-level rewards, improved reuse of partial computations (e.g., KV caches), and greater sample efficiency under memory and budget constraints. We highlight two representative approaches:

- **Search-driven Tree Rollouts:** Other works leverage Monte Carlo Tree Search (MCTS) for tree-format response generation using the classic phases: initialization, selection, expansion, and backpropagation. They view a single inference as a tree rather than a single chain, and assign rewards at the node level, which can produce a more dense/fine-grained process signal. Hou et al. [2025] propose TreeRL, an on-policy tree search framework that outperforms traditional Chain-of-Thought RL (ChainRL) while substantially reducing computational overhead through more efficient search strategies. Concurrently, ToTRL [Wu et al., 2025c] introduces a Tree-of-Thought–guided training paradigm in synthetic puzzle environments, enabling emergent generalization to out-of-distribution tasks such as mathematical reasoning. Additionally, Yang et al. [2025f] integrate MCTS into training pipelines to generate rule-based, fine-grained process rewards, improving reward signal granularity and fidelity in policy optimization.

- **Shared-prefix or Segment-wise Schemes:** While these tree search methods enrich exploration and provide fine-grained rewards, their sample efficiency remains a limitation. Some works design segmented/shared prefix sampling to improve generation efficiency [Guo et al., 2025c, Hou et al., 2025, Li et al., 2025q, Yang et al., 2025f]. SPO [Guo et al., 2025c], TreeRPO [Yang et al., 2025f], TreeRL [Hou et al., 2025], FR3E [Zheng et al., 2025c], and ARPO [Dong et al., 2025b] conduct additional sampling starting from previously generated prefix. TreePO [Li et al., 2025q] implements a segment-wise tree sampling algorithm that alleviates the KV cache burden, reducing the GPU hours for training, and improving sampling efficiency.

3.3.2. Sampling Hyper-parameters

> **Takeaways**
>
> - Careful hyperparameter tuning is essential for scalable RL, as naive settings can lead to inefficiency and unstable training (e.g., entropy collapse).
>
> - Scalable RL relies on a holistic combination of strategies to balance cost and stability, such as staged context lengthening and dynamic exploration controls.

This subsection summarizes the hyperparameter adjustment strategies for sampling from recent works. Effective RL training requires a delicate balance between several competing objectives, and recent literature has focused on techniques across two primary axes: 1) managing the exploration-exploitation trade-off to ensure the model discovers and refines effective reasoning paths; 2) efficiently managing sequence length to balance reasoning depth with computational cost.

Exploration and Exploitation Dynamics. A central challenge is balancing exploration (discovering novel reasoning strategies) with exploitation (refining high-reward solutions). The primary levers for this are temperature, entropy regularization, and PPO's clipping mechanism. For temperature,

strategies vary significantly. Some works propose a dynamic approach, such as staged temperature increases (e.g., $1.40 \rightarrow 1.45 \rightarrow 1.50$ for a 4B model, $0.7 \rightarrow 1.0 \rightarrow 1.1$ for a 7B model) to gradually expand trajectory diversity as training progresses [An et al., 2025], or using a scheduler to dynamically adjust temperature to maintain a stable entropy level [Liao et al., 2025b]. A more prescriptive approach recommends tuning the training temperature to keep the post-scaling entropy around a target of 0.3, which is found to strike an optimal balance [Liu et al., 2025u, Wu et al., 2025e]. Other works simply advocate for a high, fixed temperature (e.g., 1.0 or 1.2) to encourage initial exploration, while noting it is insufficient on its own to prevent long-term entropy decline [Arora and Zanette, 2025, Liu et al., 2025i, Shrivastava et al., 2025].

Length Budgeting and Sequence Management. Nearly all works grapple with managing the length of generated responses to balance performance and cost. The most prevalent strategy is staged context lengthening [Luo et al., 2025c]. This involves starting RL with a short context window (e.g., $8k$) before progressively increasing it to $16k$, $24k$, or $32k$ in later stages [Chen et al., 2025q, Liu et al., 2025i,u, Luo et al., 2025c]. The initial short-context stage is considered essential, as it forces the model to learn more concise and token-efficient reasoning patterns [Chen et al., 2025q, Liu et al., 2025u, Luo et al., 2025c]. An alternative to training on very long contexts is to apply inference-time length extrapolation techniques like Yarn at inference time, allowing a model trained on shorter sequences to generalize to longer ones [An et al., 2025]. For handling responses that exceed the length budget, there is no consensus. Some works apply a soft, linear penalty as the response approaches the maximum length [Yu et al., 2025d] or a tunable penalty (α) directly in the reward function [Arora and Zanette, 2025]. A more nuanced, stage-dependent strategy is to filter (mask the loss of) overlong samples when the length budget is short ($8k$-$16k$) but to penalize them when the budget is large ($32k$), as filtering can become detrimental at very long contexts [Liu et al., 2025u, Wu et al., 2025e].

Across these works, effective hyperparameter adjustment emerges as the joint tuning of exploration (temperature, entropy targets, clipping), efficiency (staged length curricula), and sequence management (overlength filters, penalties, or inference-time extrapolation). These methods are directly applicable to most GRPO/PPO-style RL pipelines for LLMs.

4. Foundational Problems

Having reviewed the key components of RL pipelines for LLMs, we now turn to several foundational problems that remain central and often unresolved in the field. In this section, we articulate the core issues, present contrasting perspectives, and summarize recent progress on each open question. Specifically, we discuss challenges such as the fundamental role of RL (sharpening versus discovery) in § 4.1, the boundary between RL and SFT (generalization versus memorization) in § 4.2, the selection of model priors (weak versus strong models) in § 4.3, the effectiveness of training algorithms (tricks versus traps) in § 4.4, and the granularity of reward signals (process versus outcome) in § 4.5. By highlighting these open questions, we aim to clarify the current landscape and motivate further investigation into the foundational underpinnings of RL for LRMs.

4.1. RL's Role: Sharpening or Discovery

We begin by summarizing the two prevailing perspectives on the role of RL: Sharpening and Discovery. These perspectives appear to be in direct opposition. The Sharpening view suggests that RL does not create genuinely novel patterns, but instead refines and reweights correct responses already contained within the base model. By contrast, the Discovery view claims that RL is capable of uncovering new patterns that the base model does not acquire during pre-training and would not generate through repeated sampling.

The divergence between the Sharpening and Discovery perspectives can be understood through multiple theoretical lenses. First, from the KL divergence optimization viewpoint, SFT typically optimizes the forward KL divergence $D_{KL}(p_{data}||p_{model})$, exhibiting *mode-covering* behavior: the model attempts to cover all modes in the data distribution. In contrast, RL methods optimize the reverse KL divergence $D_{KL}(p_{model}||p_{reward})$, which exhibits *mode-seeking* behavior: concentrating probability mass on high-reward regions [Ji et al., 2024, Sun, 2024]. Recent theoretical advances have further enriched this understanding. Xiao et al. [2025b] demonstrate that RLHF can be viewed as implicit imitation learning on preference data, establishing a deep connection between RL-based alignment and behavioral cloning. Similarly, Sun [2024] frames SFT itself as a form of inverse RL, revealing that even supervised approaches implicitly involve reward modeling. These perspectives suggest that the Sharpening vs. Discovery debate may be addressing different aspects of a unified learning process: while the mode-seeking nature of RL provides a mechanism for sharpening, the implicit reward learning and compositional capabilities could enable discovery through extended training.

- Initially, DeepSeek-R1 [Guo et al., 2025a] demonstrated promising "Aha" behaviors through RLVR, inspiring lightweight reproductions such as TinyZero [Pan et al., 2025c], which reported similar phenomena with simplified training recipes and minimal code. Domain-specific adaptations soon followed, including Logic-RL [Xie et al., 2025c], which showcased rule-based RL that fosters reflection and verification skills with transfer to mathematical reasoning.

- However, Limit-of-RLVR [Yue et al., 2025b] provides a sharpening-oriented counterargument: *Pass@K* evaluations indicate that RL enhances *Pass@1* performance, yet tends to underperform relative to base models when sampling broadly at large-*k Pass@K*. This suggests that RL predominantly narrows the search space rather than uncovering fundamentally novel solution trajectories. Concurrent debates questioned whether the observed "Aha" behaviors were genuinely induced by RL or merely latent capabilities already embedded during pre-training [Liu et al., 2025s, Setlur et al., 2025]. Mechanistic analyses further argued that RL gains often arise from entropy shaping or reward proxies. For instance, high-entropy "forking" tokens appear to dominate improvements [Wang et al., 2025l]; maximizing model confidence (RENT) and TTRL enhance reasoning without relying on external rewards [Prabhudesai et al., 2025, Zuo et al., 2025b]; and even spurious or random reward signals can shift Qwen models [Shao et al., 2025], implying that RL often surfaces pre-trained reasoning features rather than learning entirely new ones. A parallel line of work frames test-time search and compute as a meta-RL problem, proposing MRT to densify progress signals and yield better scaling of "thinking time" than outcome-only RL [Qu et al., 2025b]. Data-efficiency studies have also shown that even extreme cases such as *1-shot* RLVR can substantially improve mathematical reasoning, again aligning with the sharpening view of eliciting latent capabilities [Wang et al., 2025q]. Complementing these perspectives, a systematic study of exploration in RLVR [Deng et al., 2025a] formalizes *Pass@K* as a measure of exploration boundaries and uncovers nuanced entropy–performance trade-offs across training, instance, and token levels, thereby situating the sharpening view within a unified analytic framework. Recently, Shenfeld et al. [2025] introduce the principle of "RL's Razor," demonstrating that online RL preserves prior knowledge significantly better than supervised fine-tuning. They show that RL's advantages stem from its ability to maintain existing capabilities while adapting to new tasks, rather than discovering entirely novel behaviors.

- Recently, however, several works have reopened the case for discovery. ProRL [Liu et al., 2025i] reports that sufficiently prolonged and stabilized RL can extend a base model's reasoning frontier, improving both *Pass@1* and *Pass@K*. Continued scaling evidence is provided by ProRL v2 [Liu et al., 2025i], which incorporates engineering advances and demonstrates stronger results. Meanwhile, critiques of *Pass@K* metrics have led to alternatives such as *CoT-Pass@k*, supported by theoretical

arguments that RLVR implicitly incentivizes correct reasoning paths rather than merely rewarding lucky endpoints [Wen et al., 2025c]. Complementary approaches sustain RLVR's benefits by employing self-play problem synthesis to preserve entropy and enhance *Pass@K* [Liang et al., 2025c], or by directly optimizing *Pass@K* through novel policy objectives [Chen et al., 2025v, Walder and Karkhanis, 2025]. Yuan et al. [2025c] further provide compelling evidence for the discovery view by demonstrating that LLMs can learn new skills in RL through the composition of existing capabilities, suggesting that RL enables emergent behaviors beyond simple refinement of pre-existing patterns.

The apparent dichotomy between Sharpening and Discovery may be reconciled through recent theoretical advances that reveal deeper connections between different alignment paradigms. The work of Xiao et al. [2025b] shows that RLHF implicitly performs imitation learning, while Sun [2024] demonstrates that SFT can be understood as inverse RL. These insights suggest that both supervised and RL approaches are operating within a shared theoretical framework of distribution matching and reward optimization. The key distinction lies not in whether these methods can discover new capabilities, but rather in how they navigate the trade-off between exploration and exploitation. The mode-seeking property of reverse KL in RL provides a mechanism for efficient convergence to high-performance regions (Sharpening), while the implicit reward learning and sequential decision-making aspects enable the composition of existing capabilities into novel behaviors (Discovery) when given sufficient training time and appropriate regularization [Liu et al., 2025i, Yuan et al., 2025c]. This unified perspective suggests that the debate should shift from "Sharpening or Discovery" to understanding the conditions under which each phenomenon dominates.

4.2. RL vs. SFT: Generalize or Memorize

In this subsection, we discuss the roles of RL and supervised fine-tuning, focusing on the interplay between generalization and memorization. There are two primary approaches to post-training LLMs: SFT and RL. Current debates focus on two main questions: 1) Which method better enables out-of-distribution generalization? 2) Does behavior cloning via SFT set an upper bound on generalization capabilities? Recently, significant research attention has been devoted to this topic. Notably, Chu et al. [2025a] provide a direct conclusion across both textual and vision environments, stating that "SFT memorizes, RL generalizes."

Two recent studies sharpen this contrast. Huan et al. [2025] find that RL on math tasks (RL-on-math) tends to preserve, or even enhance, performance on non-math tasks and instruction following, whereas supervised fine-tuning on math (SFT-on-math) often leads to negative transfer and catastrophic forgetting. Their diagnostic analyses based on latent-space PCA and token-distribution (KL) measures, as well as those by Mukherjee et al. [2025], suggest that SFT induces representation and output drift (memorization), while RL better preserves the base-domain structure (generalization). Complementarily, Zhou et al. [2025d] dissect five math problem-solving training routes and observe that 1) continual pretraining on math text provides only modest transfer, 2) conventional *short*-CoT SFT frequently harms generalization, yet 3) *long*-CoT SFT and rule-based RL (with format/correctness rewards) expand reasoning depth and self-reflection and thus improve broader reasoning; moreover, an SFT warmup *before* RL stabilizes the policy and further boosts cross-domain transfer. These results suggest that on-policy objectives and longer, self-reflective traces foster transferable patterns that remain robust under distribution shift, whereas short-CoT SFT tends to overfit to surface patterns, mirroring the classic RL-vs.-SFT divide between generalization and memorization. There are three main research directions on this topic:

- **RL demonstrates superior generalization**: Chu et al. [2025a] show that RL outperforms SFT

in terms of Out-of-Distribution (OOD) performance, while SFT tends to memorize data on the GeneralPoints and V-IRL tasks. Previous studies [Kirk et al., 2023] have also indicated that RLHF, particularly under greater distribution shifts, can generalize more effectively than SFT, though this may come at the cost of reduced output diversity. Additionally, DeepSeek-R1 [Guo et al., 2025a] demonstrates that pure RL training can lead to the spontaneous emergence of advanced reasoning behaviors, such as reflection and verification.

- **RL is not a panacea**: The generalization ability of RL is strongly influenced by the initial data distribution and the design of verification rewards. Jin et al. [2025d] find that RL can partially mitigate overfitting; however, it remains ineffective in cases of severe overfitting or abrupt distributional shifts, as observed in OOD "24 points" and spectrum analysis tasks. The primary value of RL lies in its ability to facilitate "proper learning" [Swamy et al., 2025]. SFT can significantly improve generalization when appropriate reweighting, trust-region constraints, or dynamic rescaling are applied, and it often better prepares models for subsequent RL [Qin and Springenberg, 2025]. In practice, SFT may serve as a lower bound for sparse reward RL.

- **Unified or alternating paradigms of SFT and RL**: Yan et al. [2025a] present a framework that enhances RLVR by incorporating off-policy reasoning traces. Liu et al. [2025j] integrates SFT and RL into a single-stage target, theoretically overcoming the bottleneck of long-horizon sample complexity and empirically demonstrating superiority over using either approach alone. Fu et al. [2025c] propose a joint single-stage integration of demonstration imitation (SFT) and strategy improvement (RL) using entropy perception weights. Zhang et al. [2025o] provide theoretical evidence that in scenarios involving small models, high difficulty, or sparse successful trajectories, the traditional from SFT to RL two-stage approach may fail entirely. They address this by employing a branch rollout mechanism that begins from expert anchors to effectively link the two stages. Ma et al. [2025a] find that RL excels at consolidating and enhancing existing abilities, whereas SFT is more effective at introducing new knowledge or novel model capabilities.

However, several challenges remain unresolved. One major issue is distinguishing between genuine problem-solving ability and mere memorization of answers, while simultaneously avoiding data contamination [Satvaty et al., 2024]. There is still a lack of standardized, reproducible out-of-distribution benchmarks. Additionally, RL training is highly sensitive to the initial data distribution; when SFT induces significant representation drift, the ability of RL to recover and generalize is limited [Jin et al., 2025d]. To address these challenges, there is a need to promote frameworks such as UFT [Liu et al., 2025j], SRFT [Fu et al., 2025c], and Interleaved [Ma et al., 2025a], which mechanize the integration of SFT for incorporating new knowledge with RL for amplification and robustness. Lv et al. [2025] also explore automated scheduling strategies to determine when to switch between SFT and RL and how to allocate their proportions effectively.

In conclusion, RL tends to achieve "true generalization" on verifiable tasks and under substantial distribution shifts, but it is not a panacea. Modified SFT can help bridge the remaining gaps in generalization. Consequently, best practices are converging towards unified or alternating hybrid paradigms that combine the strengths of both approaches [Chen et al., 2025c,h, Liu et al., 2025j, Lv et al., 2025, Wu et al., 2025i, Zhu et al., 2025e].

4.3. Model Prior: Weak and Strong

Recent studies have shown that RL can now perform well across a wide range of tasks when coupled with sufficiently powerful model priors and verifiable reward signals, thereby shifting the primary bottleneck from scale to the design of environments and evaluation protocols[4]. From this perspective,

---

[4]https://ysymyth.github.io/The-Second-Half/

RL serves chiefly to resharpen latent competencies already encoded during pretraining, rather than to generate novel abilities entirely from scratch.

In this subsection, we examine three key dimensions of this dependency: the comparative advantages of applying RL to base versus instruction-tuned models, the substantial variations in RL responsiveness across different model families (particularly between Qwen and Llama architectures), and the emerging strategies that can enhance RL outcomes for both weak-prior and strong-prior models, including mid-training and curriculum design.

Base vs. Instruct Models. DeepSeek-R1 first introduced a discussion on applying RL to either base models or instruct-tuned models, and it introduced two viable paradigms for post-training: 1) *R1-Zero*, which applies large-scale rule-based RL directly to a base model, yielding emergent long-horizon reasoning; and 2) *R1*, which incorporates a brief cold-start SFT stage to stabilize output format and readability prior to RL. Independently, Open-Reasoner-Zero [Hu et al., 2025b] demonstrated that a minimalist training recipe applied to base Qwen models is sufficient to scale both response length and benchmark accuracy, mirroring the training dynamics of R1-Zero. These findings suggest that base model priors are better suited to RL than those of instruct models, often producing smoother improvement trajectories than those observed when starting from heavily aligned Instruct models, where entrenched formatting and obedience priors may interfere with reward shaping.

Model Family Differences. More recent studies highlight that the choice of base model can critically shape RL outcomes. For instance, One-shot RLVR [Wang et al., 2025q] shows that introducing a single, carefully selected mathematical example can more than double MATH500 accuracy for Qwen2.5-Math-1.5B, delivering substantial average improvements across multiple benchmarks. Yet, Spurious Rewards [Shao et al., 2025] uncovers a contrasting pattern: Qwen-family models register significant gains even under random or spurious reward signals, whereas Llama and OLMo models often do not. This divergence underscores the influence of model priors and emphasizes the importance of validating RL claims across models with differing priors. The observed asymmetries suggest differences in pretraining exposure to reasoning patterns (e.g., mathematical or code CoT). Qwen models, having been extensively exposed to such distributions, tend to be more "RL-friendly", whereas comparable Llama models often exhibit brittleness when subjected to the same RLVR procedure.

Mid-training Solutions. In practice, researchers have found that this performance gap can be addressed through mid-training or annealing training strategies. In recent LLM research, annealing denotes a late-stage pre-training phase during which the learning rate decays while the data distribution is reweighted to emphasize smaller, high-quality sources such as code, mathematics, and curated QA corpora. Llama 3 [Grattafiori et al., 2024] explicitly names this phase Annealing Data, describing both a shift in the data mixture and a linear LR decay to zero. They further report that injecting small amounts of high-quality math and code at this stage substantially improves reasoning-oriented benchmarks. Earlier, MiniCPM [Hu et al., 2024b] articulated a comparable two-stage curriculum, termed stable-then-decay. During the decay (annealing) stage, they interleave SFT-style, high-quality knowledge and skill data with standard pre-training corpora, observing larger improvements than applying the same SFT only after pre-training. Similarly, OLMo 2 [OLMo et al., 2024] makes public a modern mid-training recipe: pre-training is split into a long, web-heavy stage followed by a shorter mid-training phase that up-samples high-quality and domain-specific sources, especially mathematics, while linearly decaying the LR to zero. More generally, contemporary mid-training strategies treat the joint design of learning rate schedules and data distribution switches as a first-class concern. For instance, Parmar et al. [2024] show that optimal continued-pretraining requires: 1) a two-distribution curriculum that emphasizes the target capabilities during the late stage, and 2 an annealed, non-rewarmed LR schedule where the timing of the distribution switch is determined by the LR fraction rather than a fixed token count. A recent systematic study extends this line of work,

demonstrating that a stable-then-decay mid-training curriculum that injects high-quality mathematics and chain-of-thought QA corpora makes Llama models substantially more scalable under RL-based fine-tuning, effectively narrowing the performance gap with Qwen models [Wang et al., 2025t]. Taken together, these findings suggest a practical recipe for weak-prior model families: strengthen reasoning priors through mid-training, and subsequently apply RLVR.

Strong Model Improvements. While many replications favor base models, there is mounting evidence that RL can further improve strong distilled/Instruct models when curriculum, verification, and length control are carefully designed. For example, AceReason-Nemotron [Chen et al., 2025q] reports consistent gains from math-first then code-only RL atop distilled Qwen models, with analyses showing improvements in both *Pass@1* and *Pass@K* regimes. These findings nuance a simplistic "base-only" narrative: with the right constraints, Instruct/distilled starts can also benefit, but optimization is less forgiving. A parallel line evaluates the controllability of reasoning models. MathIF [Fu et al., 2025a] highlights a systematic tension: scaling up reasoning capabilities frequently undermines instruction-following performance, particularly in the context of long-form outputs. Complementary evidence shows that explicit CoT prompting can reduce instruction-following accuracy and proposes selective-reasoning mitigations [Li et al., 2025k]. Together, these works motivate multi-objective training (format, brevity, obedience) alongside correctness/verifiability in RL.

We can summarize how model priors fundamentally shape RL outcomes in LLM training from three perspectives: 1) Base models consistently outperform instruct-tuned models as RL starting points, with DeepSeek-R1 and Open-Reasoner-Zero demonstrating emergent reasoning from minimal recipes; 2) Model families exhibit asymmetric RL responsiveness: Qwen models show gains even under spurious rewards while Llama/OLMo models require careful mid-training with annealed learning rates and high-quality math/code data injection; 3) Strong distilled models can benefit from RL but demand more sophisticated curriculum design and multi-objective optimization.

As RL increasingly serves to resharpen latent pretraining competencies rather than create novel abilities, the focus shifts toward optimizing the pretraining-to-RL pipeline holistically rather than treating these stages independently.

4.4. Training Recipes: Tricks or Traps

RL training for large models has primarily evolved from the PPO [Schulman et al., 2017b] series, maintaining stability through a variety of engineering techniques [Huang et al., 2022] such as trimming, baseline correction, normalization, and KL regularization. In the context of RL for LLM reasoning, DeepSeek-Math and DeepSeek-R1 introduce critic-free GRPO [Shao et al., 2024], which simplifies the training process by reducing complexity. Despite these advances, challenges related to training stability and efficiency persist, motivating a range of new methods, including dynamic sampling, various importance sampling ratios, and multi-level normalization.

A more widely adopted technique to boost exploration is to use decoupled PPO clipping ("Clip-Higher"), where the upper clipping bound is set higher than the lower one (e.g., $\epsilon_{\text{low}} = 0.2, \epsilon_{\text{high}} = 0.28$) to allow the probabilities of unlikely but potentially useful tokens to increase more freely [An et al., 2025, Liu et al., 2025i, Yu et al., 2025d].

- **Minimalism in Data and Sampling**: Xiong et al. [2025a] decompose GRPO and finds that the largest performance gains come from discarding all incorrect samples, rather than relying on complex reward normalization techniques. They propose that methods like RAFT [Dong et al., 2023] or "Reinforce-Rej" [Liu et al., 2023a] can achieve stability and KL efficiency comparable to GRPO/PPO using much simpler mechanisms. DAPO [Yu et al., 2025d] systematizes "dynamic

sampling + decoupled pruning" into a reproducible large-scale approach, and incorporates decoupled PPO clipping ("Clip-Higher") where the upper clipping bound is set higher than the lower one (e.g., $\epsilon_{low} = 0.2$, $\epsilon_{high} = 0.28$) to allow the probabilities of unlikely but potentially useful tokens to increase more freely, demonstrating state-of-the-art results on strong baselines for the AIME24 benchmark. Similarly, GRESO [Zheng et al., 2025b] shows that pre-filtering can speed up rollout time by 2.4× and overall training by 2.0× with minimal loss in performance.

- **Structural Modification of the Objective Function**: GSPO [Zheng et al., 2025a] shifts ratio and cropping operations to the sequence level, resulting in improved stability and efficiency over GRPO, especially for stable RL training of Mixture-of-Experts (MoE) models. S-GRPO [Dai et al., 2025a] further reduces redundant reasoning, mitigating the tendency for longer and unnecessary reasoning chains and shortening sequence length by 35–61% across multiple benchmarks, with slight improvements in accuracy.

- **The Struggle Between De-biasing and Normalization**: Dr. GRPO [Liu et al., 2025t] identifies a key deviation in GRPO where "the longer it's wrong, the more wrong it gets," and introduces minor algorithmic modifications to improve token efficiency. At the same time, other studies (e.g., BNPO [Xiao et al., 2025a]) revisit the importance of reward normalization from an adaptive distribution perspective, proposing new normalization families. The evidence from these two camps is contradictory, indicating that viewing normalization as a universal solution may be misleading.

Liu et al. [2025v] present a recent review with unified evaluation, incorporating common techniques into a single open-source framework [Wang et al., 2025m] to enable isolated and reproducible experiments. This work provides a roadmap outlining "which techniques are effective under what settings" and demonstrates that a minimalist combination of methods can outperform GRPO and DAPO across multiple configurations. Crucially, it highlights the field's most pressing challenges: inconsistent experimental settings, incomplete reporting, and conflicting conclusions. This constitutes a fundamental limitation in the current application of RL within the research community. In summary, while practical "tricks" are valuable for stabilizing RL training, the essence of "scientific training" lies in verification and scalability. Progress in the field requires unified experimental protocols, verifiable reward structures, and explicit scalability–performance–cost curves [Nimmaturi et al., 2025] to show that a method remains effective as it scales, rather than only at specific data or models.

4.5. Reward Type: Process or Outcome

In standard RL, the objective of the policy is to maximize the expected cumulative reward [Sutton et al., 1998]. The "Reward is Enough" hypothesis [Bowling et al., 2023, Silver et al., 2021] further posits that appropriately designed rewards are sufficient and that maximizing returns can, in principle, give rise to all aspects of intelligence. In the context of RL for LLMs, the core challenge is how to provide meaningful rewards, such as training a reward model or verifier to score outputs and using these scores for RL or search. Common approaches include outcome rewards, which evaluate only the final result (e.g., correctness or passing individual tests), and process rewards, which provide step-by-step scoring through dense feedback on intermediate steps [Lightman et al., 2024].

- As shown in § 3.1.1, when task answers are verifiable, outcome rewards are the simplest and most scalable for challenging mathematical and coding tasks. However, outcome-only approaches may tacitly encourage unfaithful chain-of-thought [Arcuschin et al., 2025], such as "answer first, hallucinate later," and reward speculation. Recent research [Baker et al., 2025] indicates that state-of-the-art models also exhibit unfaithful reasoning and post-hoc rationalization in real-world

scenarios. Other work has highlighted that rule-based RL is prone to reward hacking and the development of reasoning illusions [Sun et al., 2025h].

- PRMs [Zhang et al., 2025e] naturally facilitate long-chain credit assignment. Lightman et al. [2024] clearly compare the two reward approaches: for mathematical reasoning, PRMs trained with process supervision are more stable and reliable, significantly outperforming those supervised solely by results. Nevertheless, step-wise annotation is extremely costly, and quality often declines across different domains [Zhang et al., 2025t]. Relevant studies suggest that heuristic or Monte Carlo–based synthesis approaches tend to generalize poorly and introduce bias [Yin et al., 2025].

Overall, outcome rewards provide "scalable goal alignment with automated verification", while process rewards offer "interpretable dense guidance." Combining the two, for example via implicit process modeling [Cui et al., 2025a] or generative verifiers [Zhang et al., 2024a], may represent a promising future direction in reward design.

5. Training Resources

Effective RL for LLMs depends not only on algorithms and objective design, but also on the quality and structure of the underlying training resources. The selection of resources ranging from static corpora to dynamic environments and specialized RL infrastructure, profoundly influences both the stability and scalability of large-scale training. In this section, we survey the key categories of training resources leveraged in current practice. We first examine the role and limitations of static corpora as a foundation for RL (§ 5.1), then discuss the growing importance of dynamic, interactive environments that provide richer learning signals and more realistic task distributions (§ 5.2). Finally, we review the RL infrastructure that enables scalable and efficient training pipelines for LLMs (§ 5.3).

5.1. Static Corpus

Takeaways

- RL reasoning datasets are moving from large-scale raw data to higher-quality, verifiable supervision using distillation, filtering, and automated evaluation to boost sample effectiveness and process fidelity.

- Data coverage has expanded beyond single domains (math/code/STEM) to include search, tool use, and agentic tasks with traceable, plan–act–verify trajectories.

This section surveys static corpora for RL with LLMs. Data construction is shifting from "scale-first" to "quality- and verifiability-first", explicitly to support verifiable rewards (see § 3.1.1). As shown in Table 4, the dataset coverage spans four major tracks: mathematics, coding, STEM, and agentic tasks (e.g., search and tool use). All corpora are directly compatible with RLVR, enabling process-aware evaluation. These datasets support key components of the RL pipeline, including policy pretraining, reward modeling, and difficulty-aware sampling.

Math-focused RL datasets coalesce around three construction pipelines, including annotation/verification, distillation, and multi-source merging, while widely exposing intermediate reasoning traces and spanning sizes from hundreds to millions of examples. Compact, carefully curated sets such as LIMO [Ye et al., 2025d] and LIMR [Li et al., 2025o] emphasize high-quality problems with explicit process feedback; annotated/verified resources like DAPO [Yu et al., 2025d], Big-MATH [Albalak et al., 2025], and DeepMath [He et al., 2025h] deliver reliable solution trajectories suitable for reward

Table 4 | Static datasets for RL training of LLMs, including Math, Code, STEM, and Agent domains. For data acquisition methods, "Distil" and "Anno" indicate distillation and annotation, respectively. "Merge" indicates the integration of existing datasets, including difficulty and quality filtering.

Domain	Date	Name	#Sample	Format	Type	Link
Math	2025.02	DAPO	17k	Q-A	Anno	🎧 🤗
	2025.02	PRIME	481k	Q-A	Merge&Distil	🎧 🤗
	2025.02	Big-MATH	47k	Q-A	Anno	🤗
	2025.02	LIMO	800	Q-C-A	Anno	🎧 🤗
	2025.02	LIMR	1.39k	Q-A	Anno	🎧 🤗
	2025.02	DeepScaleR	40.3k	Q-C-A	Distil	🤗
	2025.02	NuminaMath 1.5	896k	Q-C-A	Anno	🎧 🤗
	2025.02	OpenReasoningZero	72k	Q-A	Merge&Distil	🎧 🤗
	2025.02	STILL-3-RL	90k	Q-A	Merge&Distil	🎧 🤗
	2025.02	OpenR1-Math	220k	Q-C-A	Distil	🎧 🤗
	2025.03	Light-R1	79.4k	Q-C-A	Merge	🤗
	2025.04	DeepMath	103k	Q-C-A	Distil&Anno	🎧 🤗
	2025.04	OpenMathReasoning	5.5M	Q-C-A	Distil	🎧 🤗
	2025.07	MiroMind-M1-RL-62K	62k	Q-A	Merge	🎧 🤗
Code	2024.12	SWE-Gym	2.4k	Q-A	Anno	🎧 🤗
	2025.01	codeforces-cots	47.8k	Q-C-A	Distil	🤗
	2025.01	SWE-Fixer	110k	Q-A	Anno	🎧 🤗
	2025.03	KodCode	268k	Q-A	Distil	🎧 🤗
	2025.03	Code-R1	12k	Q-A	Merge	🎧 🤗
	2025.04	Z1	107k	Q-C-A	Distil	🎧 🤗
	2025.04	LeetCodeDataset	2.9k	Q-A	Anno	🎧 🤗
	2025.04	OpenCodeReasoning	735k	Q-C-A	Distil	🤗
	2025.04	DeepCoder	24k	Q-A	Merge	🎧 🤗
	2025.05	rStar-Coder	592k	Q-C-A	Distil&Anno	🎧 🤗
STEM	2025.01	SCP-116K	182k	Q-C-A	Distil	🤗
	2025.02	NaturalReasoning	2.15M	Q-C-A	Distil	🤗
	2025.05	ChemCoTDataset	5k	Q-C-A	Distil	🤗
	2025.06	ReasonMed	1.11M	Q-C-A	Distil	🎧 🤗
	2025.07	MegaScience	2.25M	Q-C-A	Merge&Distil	🤗
	2025.09	SSMR-Bench	16k	Q-A	Anno	🎧 🤗
Agent	2025.03	Search-R1	221K	Q-A	Anno	🤗
	2025.03	ToRL	28K	Q-A	Merge	🎧
	2025.03	ToolRL	4K	Q-C-A	Distil	🎧
	2025.05	ZeroSearch	170K	Q-A	Anno	🎧 🤗
	2025.07	WebShaper	0.5K	Q-A	Anno	🤗
	2025.08	MicroThinker	67.2K	Q-A	Anno	🤗
	2025.08	ASearcher	70K	Q-A	Anno	🤗
Mix	2025.01	dolphin-r1	300k	Q-C-A	Distil	🤗
	2025.02	SYNTHETIC-1/2	2M/156K	Q-C-A	Distil	🤗 🤗
	2025.04	SkyWork OR1	14k	Q-A	Merge	🎧 🤗
	2025.05	Llama-Nemotron-PT	30M	Q-C-A	Distil	🤗
	2025.06	AM-DS-R1-0528-Distilled	2.6M	Q-C-A	Distil	🎧 🤗
	2025.06	guru-RL-92k	91.9k	Q-A	Distil	🤗

modeling and value alignment; at larger scale, NuminaMath 1.5 [Li et al., 2024b] extends process-rich samples; distillation-centric corpora including DeepScaleR [Luo et al., 2025c], OpenR1-Math [Hugging Face, 2025], and OpenMathReasoning [Moshkov et al., 2025] inherit strong-teacher or "R1-style" long-chain reasoning, supporting policy pretraining and RL-stage selection; merge-and-distill collections such as PRIME [Cui et al., 2025a], OpenReasoningZero [Hu et al., 2025b], and STILL-3-RL [Chen et al., 2025u] integrate open problems with self-generated candidates, offering difficulty stratification

and high-quality filtering signals; community-leaning releases like Light-R1 [Wen et al., 2025b] and MiroMind-M1-RL-62K [Li et al., 2025m] package lightweight, RL-ready formats for rapid iteration under compute constraints. Collectively, these resources span basic computation to competition-level problems and provide both final answers and measurable intermediate steps, enabling scalable policy learning, reward modeling, and process-based reinforcement.

Code-oriented RL datasets primarily fall into three categories: program repair/editing, algorithmic competition problems, and general code synthesis with reasoning. These datasets typically provide executable unit tests and intermediate execution traces, facilitating reward shaping and process-level evaluation. Interactive, test-driven resources such as SWE-Gym [Pan et al., 2024] target fine-grained editing policies; human-verified repair pairs like SWE-Fixer [Xie et al., 2025a] and LeetCodeDataset [Xia et al., 2025c] support value alignment and reward modeling. For competition-style and algorithmic reasoning, codeforces-cots [Penedo et al., 2025], Z1 [Yu et al., 2025f], and OpenCodeReasoning [Ahmad et al., 2025] emphasize long-chain trajectories and difficulty stratification. In large-scale, "R1-style" distillation for general code generation, KodCode [Xu et al., 2025h] and rStar-Coder [Liu et al., 2025p] provide process-rich samples that aid policy pretraining and RL-stage selection. Lightweight, merge-centric releases such as Code-R1 [Liu and Zhang, 2025] and DeepCoder [Luo et al., 2025b] are convenient for rapid iteration under compute constraints. Collectively, these corpora span single-function repair through competition-level problem solving, offering both automatically checkable end artifacts and stepwise plans/edits, thereby enabling scalable policy learning, reward modeling, and process-based reinforcement for code agents.

STEM-oriented RL datasets generally converge on three themes: textbook or curriculum extraction, cross-disciplinary large-scale reasoning, and domain-specialized corpora (e.g., chemistry and medicine) featuring merge-and-distill pipelines. These datasets commonly release chain-of-thought rationales and evidence-aligned signals, enabling process-level rewards. SCP-116K [Lu et al., 2025a] targets undergraduate-to-doctoral science with automatically extracted problem–solution pairs plus model-generated reasoning. NaturalReasoning [Yuan et al., 2025e] offers multi-discipline questions decontaminated from popular benchmarks with extracted reference answers. ChemCoTDataset [Li et al., 2025c] contributes chemistry-specific CoT exemplars spanning molecular editing/optimization and reaction prediction. ReasonMed [Sun et al., 2025f] provides multi-agent–distilled medical QA with multi-step CoT rationales and concise summaries. SSMR-Bench [Wang et al., 2025u] programmatically synthesizes music-theory-grounded sheet-music reasoning questions in both textual (ABC notation) and visual formats, releasing 16k training pairs per modality, and supporting evaluation as well as RL with verifiable rewards. MegaScience [Fan et al., 2025a] aggregates public scientific corpora via ablation-based selection and annotates step-by-step solutions for most constituent sets, forming a large training pool for RL on scientific reasoning.

Mixed-domain RL datasets unify math, code, and scientific reasoning through distillation-first and merge-centric pipelines, while broadly releasing chain-of-thought traces, verifier signals, and multi-trajectory candidates that enable process rewards and difficulty-aware selection. In R1-style mixtures, dolphin-r1 [Team, 2025b] blends DeepSeek-R1, Gemini-thinking, and curated chat data for general reasoning. The SYNTHETIC suite couples large-scale SFT-style traces with RL-ready multi-trace samples: SYNTHETIC-1 [Mattern et al., 2025] aggregates DeepSeek-R1 reasoning with diverse verifiers, and SYNTHETIC-2-RL [Mattern et al., 2025] provides multi-domain tasks with multiple trajectories for preference/reward learning. SkyWork OR1-RL-Data [He et al., 2025d] emphasizes verifiable math and code problems with difficulty labels, serving as a lightweight RL pool. Llama-Nemotron Post-Training [Bercovich et al., 2025] compiles instruction/R1-style data spanning math, code, STEM, general reasoning, and tool use for post-training. AM-DeepSeek-R1-0528-Distilled [a-m team, 2025] offers cross-domain distilled traces with documented quality filtering, and guru-RL-92k [Cheng et al., 2025d] curates six high-intensity reasoning domains via a five-stage pipeline

optimized for RL formats. Collectively, these corpora provide verifiable endpoints and stepwise rationales across domains, supporting scalable policy learning, reward modeling, and process-based reinforcement.

Agent-centric RL datasets concentrate on two complementary capabilities, search-as-action and tool use, while releasing verifiable process signals such as search/browse traces, evidence URLs, and tool-execution logs that enable process rewards and offline evaluation. Search-R1 [Jin et al., 2025b] builds on NQ/HotpotQA to train interleaved reasoning–search behavior. ToRL [Li et al., 2025p] scales tool-integrated RL from base models to learn when and how to invoke computational tools. ToolRL [Qian et al., 2025] studies fine-grained reward design for learning tool selection and application. ZeroSearch [Sun et al., 2025a] formulates offline information-seeking tasks that incentivize search without real web calls. WebShaper [Tao et al., 2025] synthesizes information-seeking data via an "Expander Agent", covering diverse task forms and reasoning structures with URL evidence. MicroThinker [Team, 2025f] contributes full rollout trajectories and rich tool-use logs for multi-step agents. ASearcher [Gao et al., 2025a] releases Apache-2.0-licensed training splits for long-horizon search agents with question/answer fields and source annotations. Collectively, these corpora span planning, retrieval, tool orchestration, evidence verification, and answer generation, supporting scalable policy learning, reward modeling, and process-based reinforcement for web/search and tool-using agents.

5.2. Dynamic Environment

> **Takeaways**
>
> - Static RL training datasets are increasingly insufficient for advanced and generalizable reasoning abilities.
>
> - Scalable RL for LLMs needs to turn to synthesized or generated data and interactive environments, such as various gyms and world models.

Existing static RL corpora, whether manually annotated, semi-automatically labeled, or scraped from the Web, are increasingly insufficient for training models that require more advanced and generalizable reasoning abilities. A growing number of works are now leveraging "Dynamic Environments" to jointly ensure both *scalability* and *verifiability*, two essential properties for effective model training [Wei, 2025].

Unlike traditional reasoning corpora, these dynamic environments represent a paradigm shift. They enable either the automated and limitless synthesis of data, or provide step-level, multi-turn feedback on a model's entire reasoning process. As shown in Table 5, based on the methods used for synthesis and interaction, these environments can be categorized, serving as the interaction objects for the RL process. Given our focus on resources for training, this subsection's organization of datasets and environments will exclude benchmarks intended solely for evaluation.

Rule-based Environment. Relying solely on feedback like "Exact Match" can lead models to shortcut to memorization rather than actual reasoning. To counteract this, some environments offer complex and diverse tasks that require deterministic rule-based operations as a verifier. AutoLogi [Zhu et al., 2025d] generates open-ended logic puzzles with controllable difficulty by building code that checks the correctness of logical constraints based on a fixed model output format. Logic-RL [Xie et al., 2025c] uses a scalable Knights and Knaves puzzle to create a rule-based RL environment, which generalized the reasoning capabilities of a 7B model to the mathematical domain. Projects like SynLogic [Liu et al., 2025g], Reasoning Gym [Stojanovski et al., 2025], and Enigmata [Chen et al.,

Table 5 | Dynamic RL Environments for RL Training of LLMs. Data source legend: **RD** = Read Data, **RS** = Rule-based Synthesis, **MS** = Model-based Synthesis. Scale legend: Training/Test set.

Category	Date	Name	Data Source	Interactive	Scale	Multimodal	Link
Rule-based	2025.02	AutoLogi	RD + MS	✗	2458/6739 puzzles	✗	⊙
	2025.02	Logic-RL	RS	✗	5k samples	✗	⊙
	2025.05	Reasoning Gym	RS	✗	104 tasks	✗	⊙
	2025.05	SynLogic	RS	✗	35 tasks	✗	⊙🤗
	2025.06	ProtoReasoning	RD + MS	✗	6620 samples	✗	-
	2025.06	Enigmata	RD + RS	✗	36 tasks	✗	⊙🤗
	2025.07	StepFun-Prover	RD + RS	✗	36 tasks	✗	⊙🤗
	2025.08	FTRL	RD + MS	✓	2215/200 samples	✗	⊙
Code-based	2024.07	AppWorld	RD + RS	✓	750 tasks	✗	⊙
	2025.02	AgentCPM-GUI	RD + RS	✓	55k trajectories	✓	⊙🤗
	2025.02	MLGym	RD + RS	✓	13 tasks	✗	⊙
	2025.03	ReCall	RD + MS	✓	10010 samples	✗	⊙🤗
	2025.04	R2E-Gym	RD + MS	✓	8135 cases	✗	⊙🤗
	2025.05	MLE-Dojo	RD + RS	✓	202 tasks	✓	⊙🤗
	2025.05	SWE-rebench	RD + MS	✓	21336 cases	✗	⊙🤗
	2025.05	ZeroGUI	MS	✓	-	✓	⊙🤗
	2025.06	MedAgentGym	RD	✓	72,413 cases	✗	⊙🤗
Game-based	2020.10	ALFWorld	RS	✓	6 tasks	✓	⊙
	2022.03	ScienceWorld	RS	✓	30 tasks	✗	⊙🤗
	2025.04	Cross-env-coop	RS	✓	1.16e17 cases	✗	⊙
	2025.05	lmgame-BENCH	RD + RS	✓	6 games	✓	⊙🤗
	2025.05	G1(VLM-Gym)	RD + RS	✓	4 games	✓	⊙
	2025.06	Code2Logic (GameQA)	RD + MS	✗	140k QA	✓	⊙🤗
	2025.06	Play to Generalize	RS	✓	36k samples × 2 games	✓	⊙🤗
	2025.06	KORGym	RS	✓	51 games	✓	⊙🤗
	2025.06	Optimus-3	RS	✓	6 tasks	✓	⊙🤗
	2025.08	PuzzleJAX	RS	✓	~ 900 games	✓	⊙
Model-based	2025.03	Sweet-RL	RD + MS	✓	10k/1k tasks	✗	⊙🤗
	2025.04	TextArena	RS	✓	99 games	✗	⊙
	2025.05	Absolute Zero	MS	✓	-	✗	⊙🤗
	2025.06	SwS	RD + MS	✗	40k samples	✗	⊙🤗
	2025.07	SPIRAL	RS	✓	3 games	✗	⊙🤗
	2025.08	Genie 3	MS	✓	-	✓	⊙
Ensemble-based	2025.06	InternBootcamp	RD + RS	✓	1060 tasks	✗	⊙
	2025.07	Synthetic-2	RD + MS	✓	19 tasks	✗	🤗

2025d] expand the task diversity further. They identify the key parameters that control the difficulty for each task, allowing for the unlimited generation of data across various logic-related reasoning challenges. In contrast, ProtoReasoning [He et al., 2025b] operates on the hypothesis that a model's generalization ability comes from shared abstract reasoning prototypes. It normalizes different task types into a consistent format, like Prolog questions or PDDL tasks, and then automatically verifies the model's output using an interpreter.

Code-based Environment. An important application area for LLM reasoning is software engineering and code development. A key characteristic of these environments is that models must interact with a compilable code environment during training. Therefore, how to scalably construct code-based task environments remains a significant research direction. To teach agents to use tools, ReCall [Chen et al., 2025k] leverages advanced LLMs to construct a Python-based tool interaction environment, autonomously synthesizing its own SynTool data for RL training. In the field of AutoML, MLGym [Nathani et al., 2025] was among the first to support an interactive environment for iterative experimentation and training. It isolates each task's execution environment using Docker containers. Though its tasks are largely fixed, it offers less scalability. MLE-Dojo [Qiang et al., 2025] offers more scalability as it is easier for users to integrate new tasks. In a similar vein, MedAgentGym [Xu

et al., 2025b] is an efficient and scalable interactive training environment for the medical domain. In software engineering, R2E-Gym [Jain et al., 2025] reduces the reliance on manually authored GitHub issues and test cases by programmatically generating environments directly from GitHub commit histories, integrating with OpenHands for interactive capabilities. Similarly, SWE-rebench [Badertdinov et al., 2025] extends the original static SWE-bench by proposing a scalable pipeline for constructing software engineering tasks. This pipeline includes complex, interactive tasks that simulate real-world software development scenarios, ensuring data freshness and avoiding data contamination. In the field of computer use, AgentCPM-GUI [Zhang et al., 2025u] constructs an interactive GUI environment during the RFT phase to provide feedback on the model's actions. Similarly, AppWorld [Trivedi et al., 2024] uses an environment comprising various mobile application APIs. ZeroGUI [Yang et al., 2025b] takes this a step further by using existing advanced VLMs to construct tasks for both Ubuntu and Android. During training, a GUI agent interacts with the environment, and the feedback is then provided to the VLM to give rewards, all without the need for manual data curation.

Game-based Environment. Game environments are characterized by their clear and complex state spaces, where an AI's behavior is tightly coupled with the environment's state.This leads to a more multi-step and continuous interaction process compared to the environments mentioned previously, and such environments naturally support dense rewards in § 3.1.3, making RL training more efficient and stable. Early works on interactive environments for training agents, such as ALFWorld [Shridhar et al., 2020] and ScienceWorld [Wang et al., 2022], remain influential in the agent planning field. Code2Logic [Tong et al., 2025b] utilized game code and Q&A templates to automatically generate multimodal reasoning data, resulting in the GameQA dataset. This dataset is not only scalable but also tests a model's multimodal reasoning capabilities with graduated difficulty. lmgame-Bench [Hu et al., 2025c], in a different approach, directly selects classic games and interacts with an LLM via a unified API. The game environment updates its state and provides a reward based on the LLM's action, which the LLM then uses to adjust its strategy. Similarly, Play to Generalize [Xie et al., 2025d] used a simple, scalable game environment for RL to train a 7B-parameter MLLM. The research found that the reasoning skills acquired by the model could generalize to unseen games and multidisciplinary reasoning tasks. The work G1 [Chen et al., 2025g] introduced the VLM-Gym, an RL environment that supports the parallel execution of multiple game states, facilitating large-scale training. KORGym [Shi et al., 2025a] further expands the number of supported simple games, offering interactive and difficulty-configurable RL environments. PuzzleJAX [Earle et al., 2025] takes a different approach by accelerating games generated from the PuzzleScript language using JAX. This not only speeds up the game environment to support RL-based training but also provides access to a community of game developers with a source of unlimited games. To learn general cooperative skills, Cross-environment Cooperation [Jha et al., 2025] leverages the game Overcooked and maximizes environmental diversity within a self-play framework. For more complex, high-degree-of-freedom games like Minecraft, the Optimus series of work [Li et al., 2025t] leverages knowledge graphs to interact with the game environment, constructing data to evaluate a model's long-term planning ability.

Model-based Environment. This paradigm facilitates the creation of highly flexible and diverse RL environments through model-to-model interaction or self-play. SwS [Liang et al., 2025b] utilizes a model's failed training cases to abstract key concepts and generate new problems, thus enhancing its reasoning abilities in a targeted manner. SPIRAL [Liu et al., 2025a] uses three zero-sum games for self-play to prevent overfitting to a static policy. For model-to-model interaction, Sweet-RL [Zhou et al., 2025g] uses a prover-verifier-like training framework, where an agent interacts and collaborates with an LLM-based human simulator to solve front-end design and back-end programming tasks. TextArena [Guertler et al., 2025] proposes using adversarial text games combined with a ranking system, which overcomes the bottleneck of human scoring by allowing models to interact directly to relatively measure their abilities. Absolute Zero [Zhao et al., 2025a] goes a step further by completely

moving away from human-defined evaluation tasks, utilizing three reasoning modes for a model to autonomously generate its own tasks and improve its reasoning capabilities through self-evolution. In the visual domain, Genie 3 [Ball et al., 2025] generates near-realistic and interactive 3D virtual environments, laying the foundation for future multimodal environment-interactive RL. While some existing world models have already enabled RL-based model training [Dedieu et al., 2025, Hafner et al., 2023, Russell et al., 2025], and we have listed works that train LRMs using model-based environments above, there is still no sufficiently scalable solution to support RL training of LRMs based on world models. The ultimate form of such dynamic environments, we posit, would be an oracle world model capable of simulating a complete, self-contained world.

Ensemble-based Environment. There are also works that involve significant engineering effort that integrate various tasks and datasets to form interactive environments and training data for RL. InternBootcamp [Li et al., 2025f] is a large-scale, extensible library of environments designed to train LRMs. It supports over 1000 general reasoning tasks across eight domains by providing difficulty-controllable generators and rule-based verifiers. A key contribution is its empirical demonstration of "Task Scaling," showing that increasing the number of training tasks significantly boosts both reasoning performance and training efficiency. Synthetic-2 [PrimeIntellect, 2025] contributes to this approach by providing a massive, open dataset of four million verified reasoning traces. These traces were collaboratively generated via a "planetary-scale, pipeline-parallel, decentralized inference run," showcasing a highly scalable method for creating verified training data for complex RL tasks.

5.3. RL Infrastructure

> **Takeaways**
>
> - Modern RL infrastructure centers on flexible pipelines and communication layers that allocate resources between agent rollout and policy training, typically implemented as wrappers over mature distributed training frameworks and inference engines.
>
> - Specialized variants (agentic workflows, multi-agent, and multimodal) commonly support asynchronous rollouts/training and standardized environment interfaces.

In this subsection, we introduce the open-source RL infrastructure that promotes the development not only in algorithmic research but also in downstream applications. We begin by presenting primary development frameworks, which mainly provide basic wrappers around LLM training and inference frameworks. Next, we introduce secondary development frameworks, which are built upon these primary frameworks and further adapted to various downstream applications, including agentic RL, coding RL, multi-agent RL, and multimodal RL, distributed RL, and others. We compare these open-source RL frameworks in Table 6 and introduce the main frameworks below..

Primary Development. Current RL infrastructure relies heavily on mature training frameworks and inference engines designed for LLMs. Frameworks such as DeepSpeed [Rasley et al., 2020], Megatron [Shoeybi et al., 2019], and Fully Sharded Data Parallel (FSDP) [Zhao et al., 2023b] are optimized for both pre-training and post-training of LLMs. In terms of inference, vLLM [Kwon et al., 2023] and SGLang[5] are tailored for efficient inference, incorporating advanced schedulers and flash attention mechanisms. These optimizations enable significantly faster inference compared to direct forward computation on PyTorch models. Many open-source RL frameworks are built upon plug-and-play training and inference frameworks, most of which are implemented on distributed computing engines such as Ray[6]. Here, we review RL frameworks that are directly developed based on the

---

[5]https://github.com/sgl-project/sglang
[6]https://github.com/ray-project/ray

Table 6 | Open-source RL infrastructure for LLM post-training. Status legend: ✓ = native, ✗ = unsupported, **P** = partial.

Date	Framework	Runtime				Serving		Training		
		Async	Agents	Multi-Agents	Multimodal	vLLM	SGLang	DeepSpeed	Megatron	FSDP
Primary development										
2020.03	TRL	✗	✗	✗	P	✓	✗	✓	✗	✓
2023.11	OpenRLHF	✓	✓	✗	✗	✓	✗	✓	✗	✗
2024.11	veRL	✓	✓	✗	P	✓	✓	✗	✓	✓
2025.03	AReaL	✓	✓	✗	P	✓	✓	✓	✓	✓
2025.05	NeMo-RL	P	P	✗	✓	✓	✗	✗	✓	✓
2025.05	ROLL	✓	✓	✗	✓	✓	✓	✓	✓	✗
2025.07	slime	✓	P	✗	✗	✗	✓	✗	✓	✗
Secondary development										
2025.02	rllm	P	✓	✗	✗	✓	✓	✗	✗	✓
2025.02	VLM-R1	✗	✗	✗	✓	✓	✗	✓	✗	✗
2025.03	EasyR1	✗	✗	✗	✓	✓	✗	✗	✗	✓
2025.03	verifiers	✓	✓	✗	✗	✓	✗	✓	✗	✓
2025.05	prime-rl	✓	✗	✗	✗	✓	✗	✗	✗	✓
2025.05	MARTI	P	✓	✓	✗	✓	✗	✓	✗	✗
2025.05	RL-Factory	✓	✓	✗	✓	✓	✓	✓	✓	✓
2025.06	verl-agent	✓	✓	✗	✓	✓	✓	✓	✓	✓
2025.08	agent-lightning	✓	✓	P	✗	✓	✗	✗	✓	✓

aforementioned backbone training and inference frameworks.

- **TRL** [von Werra et al., 2020]: TRL focuses on trainer-centric post-training with SFT, PPO/GRPO, DPO, and a dedicated RewardTrainer (plus recent online variants), rather than a bespoke distributed runtime. It integrates vLLM for online methods (server or colocated modes) but does not natively target SGLang or TensorRT-LLM. Scaling is delegated to accelerate, which natively supports DDP, DeepSpeed ZeRO, and FSDP; Megatron is not a backend. Reward modeling is supported out-of-the-box through the RewardTrainer, and the library provides clear APIs for GRPO/DPO/online rollouts.

- **OpenRLHF** [Hu et al., 2024a]: OpenRLHF provides distributed implementations of PPO, GRPO, REINFORCE++ (and its baseline variant) and RLOO, and also includes preference-learning baselines such as DPO/IPO/cDPO and KTO. Its runtime supports both asynchronous pipeline RLHF and asynchronous agentic RL modes, exposing a class-based agent API for multi-turn settings. For serving, OpenRLHF integrates tightly with vLLM for high-throughput rollouts. Training is organized around DeepSpeed ZeRO-3 with Auto Tensor Parallelism (AutoTP), without requiring Megatron or FSDP. The framework ships recipes for RMs and PRMs training and integrates PRM signals into rollouts.

- **Verl** [Sheng et al., 2025]: Verl offers one of the broadest algorithm menus (PPO, GRPO, GSPO, ReMax, REINFORCE++, RLOO, PRIME, DAPO/DrGRPO, and more) together with multi-turn training and tool use. Its runtime is centered on the HybridFlow controller and adds agentic RL rollout and prototypes for disaggregated asynchronous training (with "Async and off-policy architecture" on the public roadmap). Verl supports vLLM and SGLang for serving, and provides both FSDP and Megatron-LM training backends. Reward options include model-based and function/verifiable rewards (e.g., math/coding), with multi-GPU LoRA-RL support.

- **AReaL** [Fu et al., 2025b]: AReaL targets high-throughput RL for large reasoning models with a fully asynchronous design that decouples generation from training via interruptible rollout

workers, a replay buffer, and a parallel reward service (e.g., unit-test–based code rewards), stabilized by a staleness-aware PPO objective. Empirically, the system reports up to 2.77× training speedups at matched or better final accuracy on math/code benchmarks and scales near-linearly to 512 GPUs. The open-source stack emphasizes SGLang-based rollout serving and Ray launchers for single-node to ~1K-GPU clusters, with PyTorch FSDP as the main training backend (Megatron also available); the newer "AReaL-lite" adds an algorithm-first API with GRPO examples and support for multi-turn agentic RL/RLVR workflows.

- **NeMo-RL** [NVIDIA-NeMo, 2025]: NVIDIA's NeMo stack now exposes a dedicated "NeMo RL" library and the earlier NeMo-Aligner toolkit for alignment. Algorithmically, NeMo covers SFT and preference training (DPO/RPO/IPO/REINFORCE) as well as full RLHF with PPO and GRPO, including multi-turn variants. The runtime emphasizes scalable, production-oriented orchestration and extensive parallelism; training is built on Megatron Core (tensor/data/pipeline/expert parallelism) for 100B-scale models and multi-node clusters. For serving, the NeMo framework documents deployment with TensorRT-LLM and vLLM. Reward-model training is first-class in the RLHF tutorials, with end-to-end pipelines from RM fitting to PPO.

- **ROLL** [Wang et al., 2025m]: ROLL targets large-scale RL for LLMs with GRPO/PPO/REIN-FORCE++ and additional recipes (e.g., TOPR/RAFT++/GSPO), and explicitly supports asynchronous training and agentic RL pipelines. The runtime follows a Ray-based multi-role design and integrates SGLang and vLLM for rollout serving. Training is built primarily around Megatron-Core, with FSDP2 listed on the public roadmap; DeepSpeed is acknowledged as a dependency. Reward handling is modular via Reward Workers (e.g., verifiers, sandbox tools, LLM-as-judge) and pluggable environments. A technical report details the system and scaling considerations.

- **slime** [THUDM, 2025]: Slime is positioned as an SGLang-native post-training framework for RL scaling, connecting SGLang on the rollout side with Megatron on the training side. It emphasizes infrastructure over algorithm breadth, but ships examples for dense and MoE models, and includes multi-turn + tool-calling ("Search-R1 lite"). The runtime supports asynchronous training and agentic workflows; serving is first-class via SGLang. Training uses Megatron-LM with Ray for cluster launch; reward modeling per se is not the primary focus, although verifier/"reward" signals can be produced on the rollout plane.

Secondary Development. In this part, we introduce several representative frameworks that are built upon primary development frameworks and extend their features to support a broader range of downstream applications. We primarily focus on frameworks for agentic RL, multimodal RL, and multi-agent RL. Although some primary frameworks already offer partial support for these areas, we highlight specialized frameworks designed for specific domain studies.

- **Agentic RL**: This area focuses on training LLMs to utilize external tools in a variety of scenarios, such as search engines [Jin et al., 2025b], Python interpreters [Feng et al., 2025a], web browsers [Li et al., 2025e], and more. Primary frameworks like veRL [Sheng et al., 2025] and AReaL [Fu et al., 2025b] have been updated or specifically designed to support these capabilities. A core feature of agentic RL is asynchronous generation and training, which significantly reduces computational time during long-term interactions between LLMs and external environments. The secondary frameworks are mostly built upon veRL to integrate additional tools and environments, and their new features are gradually incorporated back into veRL. More details about Agentic RL will be discussed in § 6.1 and 6.2.

- **Multimodal RL**: Although the primary development frameworks were originally designed for training language models, they are typically based on transformers, which support both inference

and training of vision language models. The main challenges in this area involve data processing and loss function design. Notable frameworks such as VLM-R1 [Shen et al., 2025a] and EasyR1 [Zheng et al., 2025d] have been developed for training vision-language models based on veRL. For multimodal generation, certain frameworks have been specifically developed for RL training of diffusion-based models, such as DanceGRPO [Xue et al., 2025]. However, these approaches are beyond the scope of this paper, and readers may refer to recent RL surveys focused on vision models for further details [Wu et al., 2025h]. More details about Multimodal RL will be discussed in § 6.3.

- **Multi-Agent RL**: Frameworks for agentic RL primarily focus on implementing dynamic workflows for asynchronous rollouts and training. While most of these frameworks are still limited to single-agent applications, LLM-based MARL remains an area under active exploration. Zhang et al. [2025d] propose the first high-performance, open-source framework for LLM-based multi-agent reinforced training and inference, enabling centralized interactions and distributed policy training. In addition, recent frameworks such as Agent-Lightning [Luo et al., 2025e] have implemented disentanglement of training and inference, making it easier to support multi-agent training. More details about Multi-Agent RL will be discussed in § 6.4.

6. Applications

Advancements in RL for LLMs are best understood through their practical impact across a variety of domains. In this section, we review recent progress and challenges associated with applying RL-trained language models to real-world tasks. We highlight how RL-driven methods have improved capabilities in coding tasks (§ 6.1), enabled more autonomous and adaptive agentic behaviors (§ 6.2), and extended LLMs to multimodal reasoning across text, vision, and beyond (§ 6.3). Further, we discuss applications in multi-agent systems (§ 6.4), robotics (§ 6.5), and medicine (§ 6.6), illustrating both the broad potential and unique requirements of each area. We provide the overall taxonomy of applications along with corresponding related works in Figure 6.

6.1. Coding Tasks

> **Takeaways**
>
> - RL has advanced LLMs' reasoning and code generation in competitive programming and domain-specific tasks, driving progress toward agentic, closed-loop coding.
>
> - However, scalability, cross-task generalization, and robust automation in large-scale software settings remain open challenges.

Recently, numerous studies have demonstrated that RL offers significant advantages in verifiable tasks. Given the inherent verifiability and practical importance of coding tasks, RL has become a core approach for improving code reasoning and continues to attract substantial attention. To systematically review the field, we categorize existing research into three directions: code generation, software engineering assistance, and agentic coding, based on task complexity and developmental trend, from simpler verifiable tasks toward more complex, autonomous agentic coding.

Code Generation. The primary objective of this direction is to generate correct and executable code. Research focuses on using RL to adjust LLM generation distributions to meet the requirements of diverse coding tasks. Following the demonstration of RL's potential for complex reasoning in DeepSeek-R1, an increasing number of studies have applied RL to code generation.
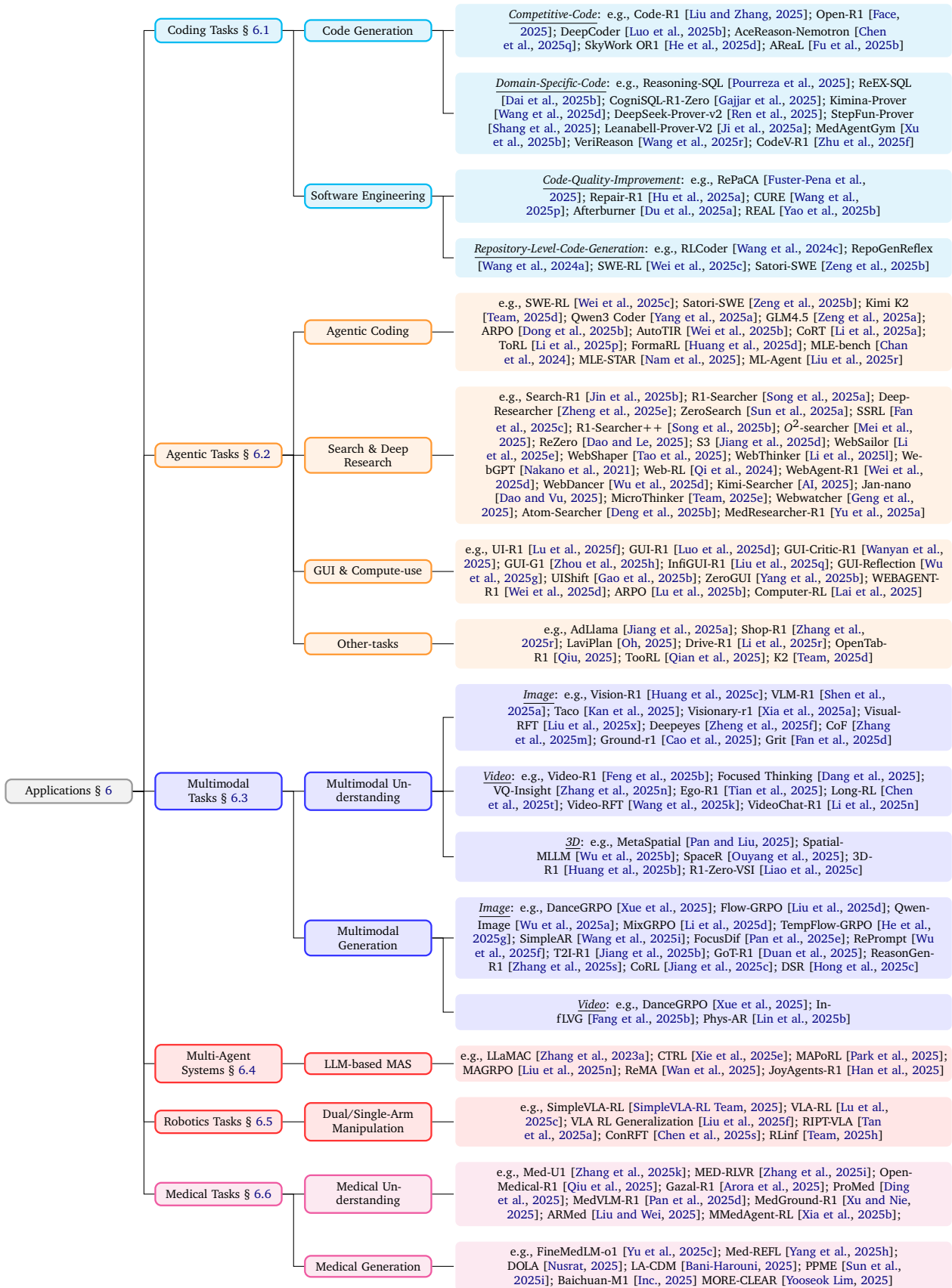
Applications § 6

Coding Tasks § 6.1

- **Code Generation**
  - *Competitive-Code*: e.g., Code-R1 [Liu and Zhang, 2025]; Open-R1 [Face, 2025]; DeepCoder [Luo et al., 2025b]; AceReason-Nemotron [Chen et al., 2025q]; SkyWork OR1 [He et al., 2025d]; AReaL [Fu et al., 2025b]
  - *Domain-Specific-Code*: e.g., Reasoning-SQL [Pourreza et al., 2025]; ReEX-SQL [Dai et al., 2025b]; CogniSQL-R1-Zero [Gajjar et al., 2025]; Kimina-Prover [Wang et al., 2025d]; DeepSeek-Prover-v2 [Ren et al., 2025]; StepFun-Prover [Shang et al., 2025]; Leanabell-Prover-V2 [Ji et al., 2025a]; MedAgentGym [Xu et al., 2025b]; VeriReason [Wang et al., 2025r]; CodeV-R1 [Zhu et al., 2025f]
- **Software Engineering**
  - *Code-Quality-Improvement*: e.g., RePaCA [Fuster-Pena et al., 2025]; Repair-R1 [Hu et al., 2025a]; CURE [Wang et al., 2025p]; Afterburner [Du et al., 2025a]; REAL [Yao et al., 2025b]
  - *Repository-Level-Code-Generation*: e.g., RLCoder [Wang et al., 2024c]; RepoGenReflex [Wang et al., 2024a]; SWE-RL [Wei et al., 2025c]; Satori-SWE [Zeng et al., 2025b]

Agentic Tasks § 6.2

- **Agentic Coding**
  - e.g., SWE-RL [Wei et al., 2025c]; Satori-SWE [Zeng et al., 2025b]; Kimi K2 [Team, 2025d]; Qwen3 Coder [Yang et al., 2025a]; GLM4.5 [Zeng et al., 2025a]; ARPO [Dong et al., 2025b]; AutoTIR [Wei et al., 2025b]; CoRT [Li et al., 2025a]; ToRL [Li et al., 2025p]; FormaRL [Huang et al., 2025d]; MLE-bench [Chan et al., 2024]; MLE-STAR [Nam et al., 2025]; ML-Agent [Liu et al., 2025r]
- **Search & Deep Research**
  - e.g., Search-R1 [Jin et al., 2025b]; R1-Searcher [Song et al., 2025a]; Deep-Researcher [Zheng et al., 2025e]; ZeroSearch [Sun et al., 2025a]; SSRL [Fan et al., 2025c]; R1-Searcher++ [Song et al., 2025b]; O^2-searcher [Mei et al., 2025]; ReZero [Dao and Le, 2025]; S3 [Jiang et al., 2025d]; WebSailor [Li et al., 2025e]; WebShaper [Tao et al., 2025]; WebThinker [Li et al., 2025l]; We-bGPT [Nakano et al., 2021]; Web-RL [Qi et al., 2024]; WebAgent-R1 [Wei et al., 2025d]; WebDancer [Wu et al., 2025d]; Kimi-Searcher [AI, 2025]; Jan-nano [Dao and Vu, 2025]; MicroThinker [Team, 2025e]; Webwatcher [Geng et al., 2025]; Atom-Searcher [Deng et al., 2025b]; MedResearcher-R1 [Yu et al., 2025a]
- **GUI & Compute-use**
  - e.g., UI-R1 [Lu et al., 2025f]; GUI-R1 [Luo et al., 2025d]; GUI-Critic-R1 [Wanyan et al., 2025]; GUI-G1 [Zhou et al., 2025h]; InfiGUI-R1 [Liu et al., 2025q]; GUI-Reflection [Wu et al., 2025g]; UIShift [Gao et al., 2025b]; ZeroGUI [Yang et al., 2025b]; WEBAGENT-R1 [Wei et al., 2025d]; ARPO [Lu et al., 2025b]; Computer-RL [Lai et al., 2025]
- **Other-tasks**
  - e.g., AdLlama [Jiang et al., 2025a]; Shop-R1 [Zhang et al., 2025r]; LaviPlan [Oh, 2025]; Drive-R1 [Li et al., 2025r]; OpenTab-R1 [Qiu, 2025]; TooRL [Qian et al., 2025]; K2 [Team, 2025d]

Multimodal Tasks § 6.3

- **Multimodal Understanding**
  - *Image*: e.g., Vision-R1 [Huang et al., 2025c]; VLM-R1 [Shen et al., 2025a]; Taco [Kan et al., 2025]; Visionary-r1 [Xia et al., 2025a]; Visual-RFT [Liu et al., 2025x]; Deepeyes [Zheng et al., 2025f]; CoF [Zhang et al., 2025m]; Ground-r1 [Cao et al., 2025]; Grit [Fan et al., 2025d]
  - *Video*: e.g., Video-R1 [Feng et al., 2025b]; Focused Thinking [Dang et al., 2025]; VQ-Insight [Zhang et al., 2025n]; Ego-R1 [Tian et al., 2025]; Long-RL [Chen et al., 2025t]; Video-RFT [Wang et al., 2025k]; VideoChat-R1 [Li et al., 2025n]
  - *3D*: e.g., MetaSpatial [Pan and Liu, 2025]; Spatial-MLLM [Wu et al., 2025b]; SpaceR [Ouyang et al., 2025]; 3D-R1 [Huang et al., 2025b]; R1-Zero-VSI [Liao et al., 2025c]
- **Multimodal Generation**
  - *Image*: e.g., DanceGRPO [Xue et al., 2025]; Flow-GRPO [Liu et al., 2025d]; Qwen-Image [Wu et al., 2025a]; MixGRPO [Li et al., 2025d]; TempFlow-GRPO [He et al., 2025g]; SimpleAR [Wang et al., 2025i]; FocusDif [Pan et al., 2025e]; RePrompt [Wu et al., 2025f]; T2I-R1 [Jiang et al., 2025b]; GoT-R1 [Duan et al., 2025]; ReasonGen-R1 [Zhang et al., 2025s]; CoRL [Jiang et al., 2025c]; DSR [Hong et al., 2025c]
  - *Video*: e.g., DanceGRPO [Xue et al., 2025]; In-fLVG [Fang et al., 2025b]; Phys-AR [Lin et al., 2025b]

Multi-Agent Systems § 6.4

- **LLM-based MAS**
  - e.g., LLaMAC [Zhang et al., 2023a]; CTRL [Xie et al., 2025e]; MAPoRL [Park et al., 2025]; MAGRPO [Liu et al., 2025n]; ReMA [Wan et al., 2025]; JoyAgents-R1 [Han et al., 2025]

Robotics Tasks § 6.5

- **Dual/Single-Arm Manipulation**
  - e.g., SimpleVLA-RL [SimpleVLA-RL Team, 2025]; VLA-RL [Lu et al., 2025c]; VLA RL Generalization [Liu et al., 2025f]; RIPT-VLA [Tan et al., 2025a]; ConRFT [Chen et al., 2025s]; RLinf [Team, 2025h]

Medical Tasks § 6.6

- **Medical Understanding**
  - e.g., Med-U1 [Zhang et al., 2025k]; MED-RLVR [Zhang et al., 2025i]; Open-Medical-R1 [Qiu et al., 2025]; Gazal-R1 [Arora et al., 2025]; ProMed [Ding et al., 2025]; MedVLM-R1 [Pan et al., 2025d]; MedGround-R1 [Xu and Nie, 2025]; ARMed [Liu and Wei, 2025]; MMedAgent-RL [Xia et al., 2025b];
- **Medical Generation**
  - e.g., FineMedLM-o1 [Yu et al., 2025c]; Med-REFL [Yang et al., 2025h]; DOLA [Nusrat, 2025]; LA-CDM [Bani-Harouni, 2025]; PPME [Sun et al., 2025i]; Baichuan-M1 [Inc., 2025] MORE-CLEAR [Yooseok Lim, 2025]

Figure 6 | Taxonomy of applications, including research directions and representative works.

- **Competitive Programming**: Competitive programming, one of the earliest benchmarks, has inspired studies including Code-R1 [Liu and Zhang, 2025], Open-R1 [Face, 2025], DeepCoder [Luo et al., 2025b], AceReason-Nemotron [Chen et al., 2025q], SkyWork-OR1 [He et al., 2025d], and AReaL [Fu et al., 2025b], which replicate DeepSeek-R1 results in code tasks. To address RL training instabilities and slow inference, DeepCoder [Luo et al., 2025b] and SkyWork OR1 [He et al., 2025d] adopted staged RL training, progressively increasing the context length to stabilize the learning process; DeepCoder [Luo et al., 2025b] and AReaL [Fu et al., 2025b] further employed asynchronous rollouts to decouple training from inference and accelerate learning. Regarding cross-task generalization, AceReason-Nemotron [Chen et al., 2025q] observed a positive transfer effect from mathematical reasoning tasks to competitive programming.

- **Domain-Specific Code**: Due to domain-specific differences in code requirements, RL is increasingly applied to specialized tasks. In data retrieval, Reasoning-SQL [Pourreza et al., 2025], ReEX-SQL [Dai et al., 2025b], and CogniSQL-R1-Zero [Gajjar et al., 2025] applied the GRPO algorithm to Text-to-SQL tasks, achieving notable performance on corresponding benchmarks. In formal proofs, Kimina-Prover [Wang et al., 2025d] and DeepSeek-Prover-v2 [Ren et al., 2025] unified informal and formal proofs by combining natural language with Lean, while StepFun-Prover [Shang et al., 2025] developed an end-to-end tool-integrated training pipeline, and Leanabell-Prover-V2 [Ji et al., 2025a] directly optimized reasoning trajectories via multi-round verifier feedback, further advancing RL's capabilities in this field. In other domains, MedAgent-Gym [Xu et al., 2025b] provided an executable coding environment for large-scale trajectory generation to improve LLM-based medical reasoning; VeriReason [Wang et al., 2025r] and CodeV-R1 [Zhu et al., 2025f] extended RLVR to the field of electronic design automation (EDA), accelerating LLM-driven hardware design. Additionally, chart-to-code generation enables agents to process structured or visual inputs and translate them into executable code, exemplifying cross-modal domain-specific code generation [Chen et al., 2025e].

Software Engineering. Despite progress in competitive programming and domain-specific tasks, these studies often fall short of real-world software development environments. Consequently, RL research also focuses on real-world software engineering, including code repair, quality optimization, and repository-level generation.

- **Code Quality Improvement**: Automated code repair and quality improvement enhance software reliability while preserving functionality. RL significantly improves repair effectiveness and generalization, enabling models to handle unseen defects. RePaCA [Fuster-Pena et al., 2025] mitigates APR patch overfitting by guiding LLMs with chain-of-thought reasoning and GRPO-based fine-tuning, while Repair-R1 [Hu et al., 2025a] jointly optimizes test-case generation and repair, reducing reliance on post-hoc validation. Beyond bug fixing, RL enhances code efficiency, maintainability, readability, and security. CURE [Wang et al., 2025p] evolves code and unit tests via encoder-tester interactions without ground-truth supervision, and Afterburner [Du et al., 2025a] leverages execution feedback, raising *Pass@1* from 47% to 62% and surpassing human-level efficiency. REAL [Yao et al., 2025b] integrates program analysis and unit testing as hybrid rewards to improve scalability and quality, achieving high-quality code generation without human intervention.

- **Repository-Level Code Generation**: Beyond function- and snippet-level tasks, recent work explores repository-level code generation and maintenance, emphasizing consistency and maintainability across complex cross-file and cross-module dependencies. RLCoder [Wang et al., 2024c] combines Retrieval-Augmented Generation (RAG) with RL to train a retriever and improve code completion accuracy. RepoGenReflex [Wang et al., 2024a] further introduces a reflection

mechanism to evaluate generated results and provide feedback, continuously optimizing generation strategies and improving generalization. By integrating RL with automated testing and continuous integration, this approach aligns LLM optimization with real-world development processes, advancing software engineering automation.

6.2. Agentic Tasks

> **Takeaways**
>
> - Agentic RL enables advanced behaviors but faces scalability issues from high computational costs and long rollout times within environments.
>
> - Asynchronous rollouts and memory agents help reduce latency and manage context, but further progress relies on better training data.

Tool use is considered a fundamental ability of language models [Schick et al., 2023]. Recent works leverage RL to help LLMs master tools and complete more complex problems [Dong et al., 2025a, Team, 2025d]. We group them into **Coding Agent**, **Simple Search Agent**, **Browser-use Agent**, **DeepResearch**, **GUI & Computer-use Agent**, and **Other Tasks**.

Coding Agent. The integration of RL and agent paradigms has advanced code generation from single-step outputs to multi-round interactions and autonomous iteration, endowing LLMs with execution and verification capabilities for closed-loop optimization.

- **Code Agents**: A common practice is to integrate RL into code agents equipped with execution and verification capabilities, and evaluate them on realistic benchmarks such as SWE-Bench. SWE-RL [Wei et al., 2025c] applies GRPO to the patch generation–execution–correction loop, enabling continuous policy optimization and improving mathematical reasoning, general code generation, and cross-domain tasks. EvoScale (Satori-SWE) [Zeng et al., 2025b] allows agents to autonomously enhance patch quality without external verifiers. RL-enhanced models such as Kimi-K2 [Team, 2025d], Qwen3-Coder, and GLM-4.5 demonstrate stronger agentic behavior, promoting greater autonomy and scalability. These developments suggest that combining RL with agentic coding is driving a shift from "single-step generation" toward "autonomous iteration."

- **Tool-Integrated Reasoning**: Another emerging application of RL lies in Tool-Integrated Reasoning (TIR), which enhances LLMs' code reasoning capabilities by tightly coupling natural language reasoning with external tool execution environments. This approach enables models to generate, execute, and verify intermediate code or program outputs, reducing errors and improving verifiability. Representative works such as ARPO [Dong et al., 2025b], AutoTIR [Wei et al., 2025b], CoRT [Li et al., 2025a], and ToRL [Li et al., 2025p] adopt similar strategies: models are post-trained with SFT or RL (mainly GRPO or variants), and outputs are structured (e.g., <code>...</code>) to trigger tool execution, feeding results back into the reasoning loop. This tight integration provides explicit RL reward signals, guiding models to produce logically consistent outputs and iteratively refine them through verifiable computation. Additionally, autoformalization approaches such as FormaRL [Huang et al., 2025d] extend TIR to Lean-based formal proof generation by integrating compiler-based syntax checks and LLM consistency evaluation with minimal labeled data, further improving reliability and correctness.

- **Automated ML Programming**: RL shows promise in automated machine learning (AutoML), expanding code agents into ML engineering agents (MLE agents) capable of autonomous data processing, model building, and optimization. MLE-bench [Chan et al., 2024] evaluates ML

agent capabilities; MLE-STAR [Nam et al., 2025] proposes a search- and optimization-based ML engineering agent; ML-Agent [Liu et al., 2025r] shows RL-driven autonomous ML engineering.

Simple Search Agent. LLMs can be trained to function as search agents through structured prompting, multi-turn generation, and integration with either online search engines (e.g., Google) or static local corpora such as Wikipedia [Jin et al., 2025a,b, Song et al., 2025a]. However, training with online search engines often incurs substantial API costs, making this approach prohibitively expensive. To address this challenge, Sun et al. [2025a] propose simulating a search engine during the training of search-capable LLMs, significantly reducing costs while maintaining or even improving performance. Other works such as R1-Search++ [Song et al., 2025b] and SEM [Sha et al., 2025] leverage the internal knowledge of LLMs to reduce training budgets while yielding better performance. Specifically, SSRL [Fan et al., 2025c] proposes training models in fully-simulated environments that can be seamlessly adapted to real scenarios through Sim2Real Generalization. Meanwhile, diverse reward signals can be developed for specific applications. Dao and Le [2025], Mei et al. [2025] employ diversity rewards to encourage comprehensive yet accurate information gathering. Wang et al. [2025v] leverage step-level rewards to further enhance the performance of search agents. S3 [Jiang et al., 2025d] utilizes gains beyond RAG to achieve better performance with fewer data. To enhance LLMs' capabilities on more challenging queries, such as those in benchmarks like GAIA [Mialon et al., 2023] and BrowseComp [Wei et al., 2025a], WebSailor [Li et al., 2025e] constructs training data from knowledge graphs, enabling models to search and browse open web environments to solve obscure problems. WebShaper [Tao et al., 2025] introduces a formalized data construction framework aimed at improving general AI assistants' problem-solving abilities.

Browser-use Agent. Besides using search engines, other browser-user agents leverage web-browsing as well. WebGPT [Nakano et al., 2021] uses textual web description to train a model to possess the ability to browse websites. Web-RL [Qi et al., 2024] employs a curriculum strategy along with ORM to convert LLMs into web agents. DeepResearcher [Zheng et al., 2025e] leverages another LLM to serve as a summarizer when browsing to help the search process. Vattikonda et al. [2025] bootstrap to train a student model using a variety of hyperparameters for stable training and better performance. WebAgent-R1 [Wei et al., 2025d] proposes a multi-turn asynchronous GRPO to train an end-to-end web browse agent, achieving strong performance. WebDancer [Wu et al., 2025d] conducts SFT and RL to enable in-depth information seeking and multi-step reasoning by web searching and browsing. Besides, other tasks are calling for a web agent, e.g., Academic Browse [Zhou et al., 2025b].

DeepResearch Agent. DeepResearch is introduced for gathering information from various sources online to help complete real-world problems, e.g., report generation. WebThinker [Li et al., 2025l], trained with iterative DPO, leverages the long-cot abilities of LRMs, using deep web explorer along with an LLM writer to finish challenging tasks. Kimi-Searcher [AI, 2025] identifies the dilemma of multi-agent, and automatically constructs intensive tool-use data to end-to-end train a single agent model, achieving great performance on HLE [Prabhudesai et al., 2025]. Jan-nano [Dao and Vu, 2025] eliminates the need for cold-start or SFT by taking multi-stage RLVR, focusing on tool calling, answering quality, and extending response length, respectively. MicroThinker [Team, 2025e] uses SFT and DPO to train Qwen3 [Wu et al., 2025a], enhancing its performance in real-world applications. Recently, WebWatcher is proposed [Geng et al., 2025] which is a multi-modal deepresearch-model capable of using external tools and visual information to solve extremely complex problems. Atom-Searhcer [Deng et al., 2025b] leverages an LRM as a PRM to provide fine-grained reward signals during training, achieving better performance. ASearcher [Gao et al., 2025a] scales the interaction turns to more than 10 turns to elicit the reasoning capability of the deep research agent. Besides general QA tasks, MedResearcher-R1 [Yu et al., 2025a] is proposed to solve clinical questions.

GUI & Computer-use Agent. UI-R1 [Lu et al., 2025f] is the first work to apply rule-based RL to graphical user interface (GUI) tasks. It introduces a novel rule-based action reward and is optimized using a small, human-curated training set. Building on this practice, GUI-R1 [Luo et al., 2025d], GUI-Critic-R1 [Wanyan et al., 2025], and so on [Du et al., 2025b, Lin et al., 2025a], carefully design fine-grained rule-based rewards tailored to specific objectives of GUI tasks, such as action accuracy, argument correctness, and step-level status. GUI-G1 [Zhou et al., 2025h] presents an empirical analysis of prior methods, identifying issues such as length bias, difficulty bias, and susceptibility to reward hacking, and reformulates the reward normalization scheme to mitigate these limitations. Furthermore, recent studies [Gu et al., 2025, Shi et al., 2025c] have attempted to obtain feedback from online GUI environments to better simulate real-world operating conditions. GUI-Reflection [Wu et al., 2025g] and UIShift [Gao et al., 2025b] derive binary rewards based on changes of UI elements to indicate action success or failure. Liu et al. [2025q] propose a two-stage training paradigm that explicitly enhances planning and reflective reasoning capabilities. ZeroGUI [Yang et al., 2025b] introduces an automated pipeline for generating challenging tasks and estimates rewards solely based on online environmental feedback, eliminating the need for human annotation. Different from the above step-level methods, there is a growing trend towards applying end-to-end asynchronous RL frameworks to train agents for mobile [Lu et al., 2025b,d, Ye et al., 2025b], and computer [Lai et al., 2025] use, which optimize the model using only rule-based task-level completion rewards without requiring step-wise reward signals. UI-TARS [Wang et al., 2025f] learns from mistakes and adapts to unforeseen situations through iterative training and reflection tuning. UI-TARS 2 [Qin et al., 2025] features with enhanced capabilities in GUI, Game, Code and Tool Use with end-to-end RL.

Other Tasks. Beyond search and GUI agents, RL has also been successfully applied to a variety of other agentic tasks. For example, Jiang et al. [2025a] improve ad copy generation by leveraging historical performance metrics, such as click-through rates, as reward signals to guide RL-based optimization. In the e-commerce domain, Shop-R1 [Zhang et al., 2025r] introduces a composite reward function that combines internal model logits with external hierarchical feedback to better simulate human-like decision-making in shopping environments. For autonomous driving, LaviPlan [Oh, 2025] aligns perceptual vision capabilities with context-aware decision-making, enabling more robust navigation under dynamic conditions. Similarly, Drive-R1 [Li et al., 2025r] is designed to balance reasoning and planning abilities for complex driving scenarios, improving both strategic and reactive behavior. In structured data interaction, OpenTab-R1 [Qiu, 2025] employs a two-stage training framework to enhance LLMs' proficiency in table-based question answering. Furthermore, general-purpose agentic models such as those in Qian et al. [2025] and Team [2025d] demonstrate the ability to master multiple commonly used tools (e.g., calculators, APIs, and databases) to solve diverse real-world tasks, showcasing the scalability of RL in building versatile, tool-augmented agents.

6.3. Multimodal Tasks

> **Takeaways**
>
> - RL strengthens multimodal models to address challenges such as limited-data settings, long-video reasoning, and numerically or attribute-sensitive cross-modal generation.
>
> - Exploring unified RL frameworks for understanding and generation is an urgent task.

The success of RL is evident not only in language models, but also in fostering notable progress in multimodal tasks. Specific optimization has been developed to enhance capabilities such as spatial perception [Su et al., 2025e] and cross-modal controllability [Wu et al., 2025h]. In the following, we discuss RL applications in multimodal tasks in terms of understanding and generation.

Multimodal Understanding. Compared to the language scenario, multimodal understanding demands powerful spatial perception and semantic alignment cross-modalities. Recently, a surge of research has employed RL to enhance reasoning ability across images, videos, and 3D spaces, demonstrating significant improvements in understanding capability.

- **RL in Image Understanding**: Vision-R1 [Huang et al., 2025c], VLM-R1 [Shen et al., 2025a], and Visual-RFT [Liu et al., 2025x] represent the first attempt to extend the DeepSeek-R1 styled RFT from math and code domains to multimodal perception tasks. These methods mark a shift in training paradigm: moving from data scaling in SFT toward the strategic design of verifiable reward functions tailored to task-specific objectives. They achieve strong performance on several detection and grounding benchmarks, demonstrating the advanced generalization ability of Reinforced Fine-Tuning (RFT) even with limited training data. Subsequently, several visual reasoning models [Kan et al., 2025, Xia et al., 2025a] adopt a similar thinking-answer format in an attempt to learn through trial and error. These methods enhance reasoning abilities via outcome-reward-driven optimization, eliminating the need for costly step-wise supervision or CoT training data. Recently, Deepeyes [Zheng et al., 2025f], CoF [Zhang et al., 2025m], and others [Cao et al., 2025, Fan et al., 2025d, Su et al., 2025a] have extended beyond pure text-based CoT to explicit multimodal-interleaved reasoning chains. These methods attempt to iteratively identify regions of interest in images using off-the-shelf tools [Su et al., 2025d] or image generation models [Xu et al., 2025e], achieving more interpretable reasoning processes. Other methods [Chu et al., 2025b, Chung et al., 2025] implement implicit multimodal-interleaved COT by copying and routing visual tokens during the reasoning stage, which mitigates hallucinations in long text-based CoT. Despite the remarkable success, several challenges remain to be addressed: 1) Inconsistent reasoning and answering: The thinking generated by the model fails to map to the final answer. 2) Long-chain exploration collapse: As the response length increases, the model becomes fragile and prone to generating hallucinations. 3) Sensitivity to data quality: RL sample selection is crucial, as low-quality training data may lead to suboptimal performance or even negative optimization.

- **RL in Video Understanding**: Extending video understanding capacity to interpret and reason over dynamic visual content is essential for multimodal understanding. To achieve this goal, Video-R1 [Feng et al., 2025b] introduces a systematic RL framework for video Multimodal Large Language Models (MLLMs), using a temporal-aware GPRO algorithm (T-GRPO) to improve spatial-temporal reasoning. Focused Thinking [Dang et al., 2025] employs a token-weighted reward scheme that trims verbose, generic chains-of-thought and uses graded (partial-credit) rewards to enhance video reasoning. VQ-Insight [Zhang et al., 2025n] designs hierarchical rewards with general task-specific temporal learning tailored QA process over long videos. To understand human daily lives from a first-person perspective, Ego-R1 [Tian et al., 2025] trains a chain-of-tool-thought agent via RL to tackle ultra-long egocentric videos (days or weeks in length) by dynamically invoking retrieval and vision tools for stepwise reasoning. Likewise, LongVILA [Chen et al., 2025t]'s Long-RL framework builds a large LongVideo-Reason dataset and a specialized two-stage CoT-SFT and RL pipeline with sequence parallelism, enabling MLLMs to process ultra-long videos. To automate more video CoT data creation, VideoRFT [Wang et al., 2025k] uses an LLM to generate initial rationales from rich video descriptors with a VLM refinement and introduces a semantic consistency reward to align textual reasoning with visual evidence. Meanwhile, VideoChat-R1 [Li et al., 2025n] demonstrates that targeted multi-task RL fine-tuning can markedly enhance specific spatio-temporal skills without degrading general chat performance. Collectively, these studies pave the way for the development of robust and generalizable video reasoning through RL.

- **RL in 3D Understanding**: While MLLMs have made significant progress in 2D visual understanding through RL, extending their ability to visual-spatial understanding in 3D space remains a challenging frontier [Wu et al., 2025b, Yang et al., 2025c]. MetaSpatial [Pan and Liu, 2025] employs a multi-turn RL-based optimization mechanism that integrates physics-aware constraints to enhance spatial reasoning in MLLMs. Building upon GRPO [Shao et al., 2024], Spatial-MLLM [Wu et al., 2025b] and SpaceR [Ouyang et al., 2025] demonstrate that even small-scale models can close the performance gap with much larger counterparts through R1-Zero-like training [Liao et al., 2025c]. Further, RoboRefer [Zhou et al., 2025a] expand RL-based spatial reasoning to embodied settings to ground reasoning in real-world dynamics.

Multimodal Generation. The exploration of RL in LLMs has also been extended to multimodal generation. Pioneering researches on test-time scaling [Liu et al., 2025b, Ma et al., 2025b, Singhal et al., 2025] and DPO [Black et al., 2024b, Liang et al., 2025d, Liu et al., 2025k, Tong et al., 2025a, Wallace et al., 2024] have driven significant progress in aesthetic and text fidelity in image and video generation. Recently, increasing attention has been devoted to enhance reasoning capabilities in image and video generation [Guo et al., 2025f, Jiang et al., 2025b].

- **RL in Image Generation**: Diffusion models have substantially advanced visual generation [Esser et al., 2024, Liu et al., 2023b, Rombach et al., 2022], and a growing body of research incorporates RL to implicitly perform reasoning by treating the denoising steps as the CoT trajectory [Liu et al., 2025d, Pan et al., 2025b, Xue et al., 2025]. However, GRPO exhibits an inherent conflict between ordinary differential equation (ODE) sampling in diffusion models. Specifically, GRPO relies on stochastic sampling to estimate advantage, whereas ODE sampling follows a deterministic denoising trajectory, which limits the diversity of rollout samples. To address this issue, an ODE-to-SDE conversion is employed [Liu et al., 2025d, Wu et al., 2025a, Xue et al., 2025] to encourage the stochastic term in the sampling process. Considering the inefficiency of SDE, MixGRPO [Li et al., 2025d] designs mixed sampling strategies through the integration of SDE and ODE. In addition, TempFlow-GRPO [He et al., 2025g] explicitly exploits the temporal structure in the flow-based model, enabling more precise credit assignment and policy optimization. Recently, GPT-4o has demonstrated powerful text fidelity and editing consistency [OpenAI, 2024], sparking interest in the controllability of autoregressive models. Building on large-scale image–text training data, SimpleAR [Wang et al., 2025i] directly applies GRPO for post-training and achieves remarkable performance in high-resolution image generation. To strengthen adherence to fine-grained attributes such as spatial relations and numerical consistency, FocusDiff [Pan et al., 2025e] constructs paired datasets that differ only in subtle attribute variations and uses them to train the generation model. Furthermore, RePrompt [Wu et al., 2025f] incorporates an additional multimodal understanding model into the image generation framework and trains it with GRPO to refine prompts. Meanwhile, T2I-R1 [Jiang et al., 2025b], GoT-R1 [Duan et al., 2025], and ReasonGen-R1 [Zhang et al., 2025s] unify prompt refinement and image generation within a single model, leveraging GRPO for joint optimization.

- **RL in Video Generation**: Compared to image generation, extending RL to video generation poses greater challenges in terms of temporal coherence and physical realism. DanceGRPO [Xue et al., 2025] conducts post-training on HunyuanVideo [Kong et al., 2024], and uses VideoAlign [Liu et al., 2025e] to provide rewards based on video aesthetics, motion quality, and text-video consistency. Furthermore, InfLVG [Fang et al., 2025b] employs GRPO to guide token selection according to contextual relevance, thereby enabling semantically consistent and temporally coherent long video generation. In addition, Phys-AR [Lin et al., 2025b] introduces velocity and mass as verifiable rewards for ball motion scenario, substantially enhancing the physical realism of video generation.

Currently, several ULM models employ a unified framework to optimize multimodal understanding and generation simultaneously. To this end, bidirectional [Jiang et al., 2025c] and dual [Hong et al., 2025c] rewards from text to image and from image to text are proposed to enhance both the generation and understanding capabilities. For multimodal understanding, Deepeyes and CoF have attempted to employ generative models or external tools to realize multimodal CoT. For multimodal generation, using refined text as the CoT also relies on the multimodal understanding capability. Therefore, exploring unified post-training methods for multimodal understanding and generation is an urgent task for future research. From the perspective of specific-domain, code generation can serve as a bridge between text and image generation. The application of RL to facilitate models to reason over complex charts and produce structured code for domain-specific image generation [Chen et al., 2025e,f, Tan et al., 2025b] is a promising application.

6.4. Multi-Agent Systems

> **Takeaways**
>
> - It is important to improve collaboration, reasoning, and credit assignment in Multi-Agent Systems (MAS), enabling more stable and effective teamwork on complex tasks.
>
> - Key challenges remain in developing efficient collaboration and interaction mechanisms to fully unlock collective capabilities and further raise agent performance.

Currently, most of the research on RL for LLM-based reasoning predominantly centers on single models, whereas applying RL to MAS has emerged as a prominent and frontier research direction. This section begins with an overview of the fundamental concepts of traditional RL and Multi-Agent RL (MARL), highlighting their primary challenges. Furthermore, the section discusses innovative applications of LLMs in MARL, emphasizing their advantages in information sharing and credit assignment. Finally, recent advances in MAS integrating RL with LLMs are examined, with a focus on how RL can be exploited to enhance collaboration and policy optimization among agents, thereby promoting the development of multi-agent reasoning capabilities.

Traditional MARL. In recent years, as a complex distributed intelligent system, MAS have attracted widespread attention in the field of RL [Dorri et al., 2018]. Traditional MARL [Busoniu et al., 2008] primarily focuses on the interactions and joint learning of multiple agents within a shared environment to achieve global objectives. The main challenges in conventional MARL include the complexity of credit assignment, the nonstationarity of the environment, and the efficiency of communication and cooperation among agents [Canese et al., 2021]. To address these issues, researchers propose a centralized training with decentralized execution (CTDE) paradigm [Lowe et al., 2017], in which agents share global information for policy optimization during the training phase, while decision-making during execution relies solely on local observations. Based on the CTDE paradigm, researchers introduce value-based methods (such as VDN [Sunehag et al., 2017] and QMIX [Rashid et al., 2020]), policy gradient-based methods (such as MADDPG [Lowe et al., 2017]), and actor-critic methods (such as COMA [Foerster et al., 2018]). Moreover, as PPO is considered to be SOTA in traditional RL, MAPPO has also been shown to have surprising effects in some simple collaborative tasks [Yu et al., 2022]. However, as the number of agents increases and the task complexity rises, traditional MARL methods face significant challenges in terms of sample efficiency and scalability. To address this issue, scholars have considered replacing current agent with neighboring agents in the interaction with all agents (such as MF-MARL [Yang et al., 2018]), which effectively alleviates the dimensionality curse caused by the increase in the number of agents in MARL. However, it still cannot be efficiently applied to complex task scenarios that require multiple agents to collaborate simultaneously.

LLM for MARL. The rapid development of LLMs has demonstrated tremendous potential in addressing challenges within MARL. Leveraging their powerful natural language understanding and generation capabilities, LLMs can provide effective information-sharing mechanisms in MAS. For instance, in credit assignment problems of MARL, researchers utilize LLMs to design intuitive reward allocation mechanisms, thereby enhancing the accuracy and interpretability of credit assignment. Zhang et al. [2023b] significantly improve multi-agent collaboration efficiency in sparse reward scenarios by enabling the LLMs to infer each agent's intention in real time and generate the next cooperative plan. Ding et al. [2023] leverage LLMs to parse natural language task descriptions into executable entity-level sub-goals, thereby achieving reward shaping and policy sharing, which effectively alleviates the credit assignment problem in MARL. Li et al. [2023a] utilize the LLM's "theory of mind" capability, allowing agents to generate linguistic beliefs about teammates' potential strategies, thus enabling more accurate decision-making in multi-agent coordination.

RL for LLM-based MAS. In the context of integrating RL with LLMs, research on MAS based on LLMs has gradually become a hotspot. Related studies primarily focus on how to fully leverage the language understanding and generation capabilities of LLMs, while utilizing RL to achieve efficient collaboration and policy optimization among multiple agents. Frameworks such as LLaMAC and CTRL integrate LLMs with the actor-critic architecture. LLaMAC [Zhang et al., 2023a] employs a centralized LLM-Critic to provide natural language-based value feedback to multiple LLM-Actors, thereby facilitating collaborative learning among multiple agents. CTRL [Xie et al., 2025e] trains LLMs to "self-criticize" by using synthetic data, and iteratively refines model outputs through RL (such as GRPO), which can improve test-time performance without the need for human annotation.

In large-scale multi-agent collaboration scenarios, MAPoRL [Park et al., 2025] promotes efficient and transferable collaboration in multi-turn tasks by jointly training multiple LLMs and introducing reasoning-aware rewards. MAGRPO [Liu et al., 2025n] models LLM collaboration as a cooperative multi-agent RL problem, which proposes a group-level relative policy optimization mechanism that significantly enhances the quality of multi-turn joint outputs in tasks such as writing and code generation. ReMA [Wan et al., 2025] introduces dual LLM structure of high-level agent and low-level agent, which achieves synergistic enhancement of meta-thinking and reasoning abilities through alternating freezing and updating of policies. JoyAgents-R1 [Han et al., 2025] designs a joint evolutionary training process, facilitating both diversity and consistency within heterogeneous LLM teams in open-domain question answering tasks through alternating global experience replay and individual PPO updates. AlphaEvolve [Novikov et al., 2025] designs an evolutionary optimization mechanism to coordinate multi-LLM collaboration. By directly modifying code and continuously receiving evaluation feedback, the MAS enhances the capability to handle complex coding tasks. AutoAgents [Chen et al., 2023a] significantly enhance the adaptability and problem-solving capabilities of MAS in complex tasks by dynamically generating specialized agents tailored to task requirements and incorporating an observer role for reflection and improvement.

6.5. Robotics Tasks

> **Takeaways**
>
> - RL addresses data scarcity and generalization challenges in robotics by adapting LLM-style approaches to Vision-Language-Action (VLA) models.
>
> - Allowing VLAs to learn from environment interaction and simple rewards, recent RL methods (e.g., GRPO, RLOO, PPO) achieve superior performance and novel behaviors with minimal supervision.

RL in Robotics Tasks. RL has been extensively applied in robotics, primarily focusing on three domains: robot control, Vision-and-Language Navigation (VLN), and robotic manipulation tasks. Traditional RL research in robot control has reached maturity with widespread applications, like action generation with human-like robots [Peng et al., 2018], robust quadruped locomotion execution [Hwangbo et al., 2019] and dexterous hand manipulation [Chen et al., 2023b]. Similarly, VLN tasks have seen significant progress [Anderson et al., 2018, Wang et al., 2018, 2019]. However, these domains differ substantially from LLM-based RL in terms of model architecture, scale, task types, reward function design, optimization objectives, and algorithmic approaches, and thus fall outside the scope of this survey.

Robotic manipulation tasks, enabling robots to solve diverse manipulation problems in real-world environments, represent the most challenging and fundamental aspect of embodied intelligence [Firoozi et al., 2025]. These tasks demand not only a comprehensive understanding of visual and textual information and fine-grained motor control, but also physical reasoning, long-horizon planning, and logical inference capabilities. Leveraging the remarkable text and vision processing capabilities of LLMs and VLMs, several studies have explored using these models as core components combined with action modules for manipulation tasks, such as RobotBrain [Ji et al., 2025b] and RT-2 [Zitkovich et al., 2023].

Vision-Language-Action Models. Recently, Vision-Language-Action (VLA) models, which integrate VLM backbones with action modules through unified end-to-end training, have emerged as the most promising solution and become the mainstream approach for robotic manipulation [Zhong et al., 2025]. Current VLA models follow a two-stage paradigm [Sapkota et al., 2025]: pretraining on multimodal data (e.g., Open X-Embodiment [O'Neill et al., 2024]) followed by supervised fine-tuning on teleoperated robot trajectories. However, this imitation learning paradigm suffers from critical limitations: its performance heavily depends on high-quality trajectory data that is expensive and inefficient to collect, and the resulting models exhibit poor generalization to unseen scenarios. Given the architectural, scale, and methodological similarities between VLAs and LLMs [Zhong et al., 2025], adapting LLM-style RL approaches to VLA training presents a promising direction for addressing data scarcity and generalization challenges.

Applying DeepSeek-R1's RL methodology to VLAs requires addressing several challenges: 1) Unlike LLMs that complete tasks in a single round, VLAs require multi-round environment interactions to generate complete trajectories; 2) VLAs operate in continuous action spaces; 3) Traditional RL methods rely on hand-crafted process rewards, limiting scalability. Recent works including SimpleVLA-RL [SimpleVLA-RL Team, 2025], VLA-RL [Lu et al., 2025c], VLA RL Generalization [Liu et al., 2025f], RIPT-VLA [Tan et al., 2025a], and ConRFT [Chen et al., 2025s] have pioneered the application of DeepSeek-R1's methodology to VLA training.

SimpleVLA-RL [SimpleVLA-RL Team, 2025] enables VLA models to interact with environments to rollout diverse complete trajectories, employing binary success/failure rewards as supervision signals and training OpenVLA-OFT [Kim et al., 2025] using the GRPO algorithm. With just a single demonstration trajectory, this RL approach surpasses state-of-the-art VLA models like π_0 [Black et al., 2024a] on LIBERO and RobotWin2.0 benchmarks, achieving SOTA performance and outperforming advanced RDT models in real-robot experiments. In addition, as an upgraded version of π_0, $\pi_{0.5}$ [Intelligence et al., 2025] uses multimodal robot data from different scenarios and sources for heterogeneous training, allowing VLA to provide a new milestone in generalizable real-world robot operation tasks. Similar to DeepSeek-R1's "aha moments", RL-trained VLAs also discover novel behavioral patterns. VLA RL Generalization [Liu et al., 2025f] investigates RL's impact on VLA generalization capabilities, demonstrating significant improvements over SFT in unseen environments, objects, and textures, while comparing GRPO and PPO effectiveness. RIPT-VLA [Tan et al., 2025a] employs RLOO [Ah-

madian et al., 2024] for VLA RL training. RLinf [Team, 2025h] designed a flexible, scalable RL framework for VLA RL that unifies rendering, inference, and training, improving both VLA training efficiency and performance. ConRFT [Chen et al., 2025s] iteratively trains VLAs through alternating RL and SFT rounds, progressively enhancing performance through multiple iterations.

The data efficiency, improved generalization, and minimal supervision requirements of RL effectively address VLA's current challenges of data scarcity and poor generalization. By allowing VLAs to autonomously explore and learn from trial-and-error with only outcome supervision, this approach dramatically reduces implementation costs compared to complex and expensive teleoperation data collection. Moreover, RL's data efficiency eliminates the need for large-scale expensive trajectory datasets, enabling scalable VLA post-training capabilities.

However, current VLA RL research remains primarily simulation-based. While SimpleVLA-RL [SimpleVLA-RL Team, 2025] achieved real-world deployment through Sim2Real transfer [Chen et al., 2025n], few works have yet deployed physical robots to collect real-world trajectories for RL. In addition, research on VLA RL is also limited by the current development of RL in robotics, including but not limited to sample efficiency, reward sparsity, and sim2real. Key challenges include autonomous sampling on physical robots requiring multiple devices for efficiency, continuous manual resetting and annotation.

6.6. Medical Tasks

> **Takeaways**
>
> - RL for medical LLMs faces distinct challenges: verifiable tasks allow stable reward design, while non-verifiable tasks make reward definition difficult.
>
> - Verifiable tasks use SFT+RL with rule-based rewards; non-verifiable tasks leverage DPO, rubrics, curriculum RL, or offline RL, though scalability and stability remain open issues.

RL optimizations in medical LLMs typically aim to enhance reasoning and generalization ability, often adopting a two-stage pipeline of SFT followed by RL. Existing works can be broadly categorized into **verifiable problems with rule-based rewards**, and **non-verifiable problems with generative or rubric-based rewards**.

Medical Understanding. These tasks, such as multiple-choice QA, structured prediction, clinical coding, or visual grounding, allow the use of deterministic rewards, making them the most mature field for RL in medical LLMs. The typical paradigm is a two-stage pipeline of SFT followed by RL, where algorithms such as GRPO optimize models directly against correctness-based signals. For example, HuatuoGPT-o1 [Chen et al., 2024a] enhances reasoning ability by synthesizing reliable reasoning trajectory data with a medical verifier and training the model with SFT and RL. Med-U1 [Zhang et al., 2025k] employs mixed binary correctness rewards with length penalties to ensure both accuracy and format compliance, while MED-RLVR [Zhang et al., 2025i] applies verifiable rewards to MCQA, improving OOD generalization. Open-Medical-R1 [Qiu et al., 2025] demonstrates that careful data filtering improves the efficiency of RL. Gazal-R1 [Arora et al., 2025] designs a multi-component reward system that refines accuracy, format adherence, and reasoning quality through GRPO for enhanced medical reasoning. ProMed [Ding et al., 2025] shifts medical LLMs from reactive to proactive paradigms, where LLMs can ask clinically valuable questions before decision-making, using Shapley Information Gain rewards during MCTS-guided trajectory exploration and RL.

Beyond textual QA, recent models extend rule-based rewards to vision and multi-modal tasks. MedVLM-R1 [Pan et al., 2025d] employs an RL framework that incentivizes the model to discover

human-interpretable reasoning paths without using any reasoning references through format and accuracy rewards. MedGround-R1 [Xu and Nie, 2025] introduces spatial-semantic rewards, which combine spatial accuracy reward and semantic consistency reward, for the medical imaging grounding task. ARMed [Liu and Wei, 2025] addresses reward collapse in open-ended medical VQA through adaptive semantic rewards that dynamically adjust the semantic reward during training based on historical reward distributions. Liu and Li [2025] leverage rule-based format and matching rewards to guide structured JSON generation for medical visual information extraction with only 100 annotated samples. MMedAgent-RL [Xia et al., 2025b] is an RL-based multi-agent framework that enables dynamic and optimized collaboration among medical agents. MedGemma [Sellergren et al., 2025] was post-trained with RL and is further evaluated on MedXpertQA [Zuo et al., 2025a], which is an expert-level medical multi-choice benchmark and includes a subset for assessing reasoning models.

For other clinical applications, DRG-Sapphire [Wang, 2025] applies GRPO with rule-based rewards to diagnosis-related grouping. EHRMIND [Lin and Wu, 2025] combines SFT warmup and RL VR for complex clinical reasoning tasks using electronic health records (EHR) data, including medical calculations, patient trial matching, and disease diagnosis. ChestX-Reasoner [Fan et al., 2025e] incorporates process rewards from clinical reports to train the model to emulate radiologists' step-by-step reasoning. CX-Mind [Li et al., 2025j] employs SFT and RL with format, result, and process rewards to train interleaved reasoning for chest X-ray diagnostics. To enable benchmarking of code-based medical reasoning, MedAgentGym [Xu et al., 2025b] presents a benchmark for code generation of medical agents, and demonstrates that RL can improve this reasoning ability.

Medical Generation. These tasks include multi-turn clinical dialogue [Bani-Harouni, 2025], treatment planning [Nusrat, 2025], and diagnostic narratives [Yooseok Lim, 2025], which lack unique ground-truth answers. As such, rule-based rewards are not directly applicable. While DPO has been applied to improve medical LLMs on preference-aligned generation tasks [Yang et al., 2025h, Yu et al., 2025c], large-scale RL on non-verifiable tasks is emerging but remains relatively underexplored. For example, DOLA [Nusrat, 2025] integrates LLM agents with a commercial treatment planning system, incorporating a reward function that guides the trade-offs between target coverage and organ at risk sparing for optimized treatment plan generation. LA-CDM [Bani-Harouni, 2025] proposes a two-agent structure trained via a hybrid training paradigm which combined supervised fine-tuning with RL to balance diagnostic accuracy, uncertainty calibration, and decision efficiency. In diagnostic dialogue, PPME [Sun et al., 2025i] develops a plug-and-play framework using large-scale EMRs and hybrid training to enhance LLM interactive diagnostic capabilities through specialized inquiry and diagnosis models. In clinical decision support, MORE-CLEAR [Yooseok Lim, 2025] applies multi-modal offline RL to sepsis treatment policies, improving survival-predictive decision-making in MIMIC-III/IV. Baichuan-M1 [Inc., 2025] employs a three-stage RL approach: ELO (Exploratory Log-likelihood Optimization) to enhance chain-of-thought reasoning diversity, TDPO (Token-Level Direct Preference Optimization) to address length-dependent constraints, and finally PPO with reward model feedback for policy refinement.

Overall, RL in medical LLMs is well established for verifiable problems, where deterministic correctness allows for rule-based rewards and stable GRPO training. In contrast, generation-oriented tasks remain challenging: current solutions adopt rubric-based rewards, curriculum transfer, or offline RL to approximate quality signals. The scarcity of scalable RL on non-verifiable tasks highlights a critical future direction for building trustworthy, reasoning-capable medical foundation models.

7. Future Directions

While RL for LLMs has made remarkable strides, many fundamental challenges and opportunities lie ahead. This section outlines several promising directions that are poised to shape the next wave of advances in the field. We highlight the importance of continual RL for adapting to evolving data and tasks (§ 7.1), memory-based and model-based RL for enhancing reasoning capabilities (§ 7.2 and § 7.3), and emerging approaches for teaching LLMs both efficient and latent-space reasoning (§ 7.4 and § 7.5). We also discuss frontiers in leveraging RL during pre-training (§ 7.6), applying RL to diffusion-based architectures (§ 7.7), and driving scientific discovery (§ 7.8). Finally, we consider the challenges and prospects of architecture-algorithms co-design to meet the demands of ever-larger and high-efficiency intelligent models (§ 7.9). By surveying these directions, we aim to provide both a roadmap and inspiration for future research in RL for LLMs.

7.1. Continual RL for LLMs

To enhance the multi-domain performance of LLMs during RL-based post-training, the mainstream approach is to mix data from different tasks and train in a unified manner [Guo et al., 2025a, Yang et al., 2025a]. On synthetic data [Chen et al., 2025d, Liu et al., 2025g], Multi-stage RL has been shown to perform worse than training with mixed data, and even curriculum learning with increasing difficulty may not be necessary in RL [Xie et al., 2025c]. However, Chen et al. [2025d] suggest that multi-stage RL across different tasks has advantages in generalizing to difficult or unseen problems. Despite these ongoing debates of multi-stage RL's effectiveness, as the field advances toward building AI systems that must adapt to evolving data and tasks in dynamic environments, it becomes necessary to explore Continual Reinforcement Learning (CRL) for LLMs.

Similar to traditional CRL, LLMs face the fundamental challenge of balancing stability and plasticity during multi-stage RL training [Pan et al., 2025a]. Plasticity may be particularly concerning for LLMs, as widely used deep learning techniques can cause large models to perform no better than shallow networks in continual learning settings [Dohare et al., 2024]. Another challenge of CRL for LLMs lies in the entangled nature of knowledge and reasoning in LLMs, which distinguishes from traditional RL settings where tasks can be discretely defined and policies can be modularly organized, such as in game-like environments [Chevalier-Boisvert et al., 2023, Towers et al., 2024] or embodied scenarios [Todorov et al., 2012, Wołczyk et al., 2021].

Existing methodological frameworks from traditional CRL research provide a promising foundation for addressing LLM-specific requirements. Core methodological insights from traditional CRL research, including Experience Replay [Berseth et al., 2021, Li et al., 2021, Rolnick et al., 2019], Policy Reuse [Garcia and Thomas, 2019, Gaya et al., 2022], and Reward Shaping [Jiang et al., 2021, Zheng et al., 2022]. It remains a valuable research direction for developing CRL frameworks tailored to LRMs. The development of specialized CRL techniques for LLMs or LRMs will be crucial for creating more adaptive and efficient AI systems capable of lifelong learning and operating in dynamic and ever-changing environments.

7.2. Memory-based RL for LLMs

Although many works in agentic RL have explored memory mechanisms, ranging from external long-term storage and insertion [Chhikara et al., 2025, Xu et al., 2025d, Zhong et al., 2024] to internal memory processing and working-memory control [Yu et al., 2025b, Zhou et al., 2025i], most designs remain tailored to the current task with limited generalization beyond it. As Silver and Sutton [2025] emphasize, the next generation of intelligent agents will learn primarily from experience,

acquiring skills through continual interaction. In this spirit, a key direction is to transform agent memory from task-specific buffers into experience repositories that are structured, reusable, and transferable across diverse tasks, allowing memory to evolve into a foundation for broader adaptability and lifelong learning. Such an experience-centric view also aligns naturally with RL, since the data generated from the interactions between an agent and its environment provides rich experiential traces that can be utilized effectively. Moreover, although recent works have explored maintaining a shared pool of experiences to retrieve relevant strategies from past histories and adapt other agents' experiences to new task scenarios [Tang et al., 2025], this direction remains underexplored. A core challenge here is enabling agents, through RL, to automatically learn how to operate and manage memory, composing and generalizing experiential knowledge across tasks. Addressing this challenge is essential for moving toward an "experience era" where collective interaction traces become a foundation for broader agent intelligence.

7.3. Model-based RL for LLMs

A core challenge in RL lies in obtaining scalable and robust reward signals as well as meaningful state representations from the environment. Prior work has investigated the construction of world models [Luo et al., 2024, Moerland et al., 2023] to supply informative states for RL agents, and more recently, LLMs have been adopted as world models in various RL contexts [Benechehab et al., 2024, Gu et al., 2024, Hu and Shu, 2023]. In the case of RL with LLMs, especially for language agents, the ability to construct world models that accurately capture environmental states and generate reliable rewards is critical. Recent advances show that generative world models, including those enhanced by video pre-training [Assran et al., 2025, Ball et al., 2025, Bruce et al., 2024], are both practical and effective. Nevertheless, seamlessly integrating world models with RL for LLM-based agents remains an open research problem. As such, model-based RL with LLMs is emerging as a particularly promising and scalable direction for future research.

7.4. Teaching LRMs Efficient Reasoning

Inference-time scaling has improved the accuracy of LRMs on difficult tasks, but it also introduces systematic *over-thinking* (needlessly long reasoning chains for easy instances) [Chen et al., 2024b, Qu et al., 2025a, Sui et al., 2025, Yan et al., 2025b] and, under aggressive truncation, *under-thinking* (premature halting and reliance on brittle shortcuts) [Su et al., 2025b, Wang et al., 2025s]. A central challenge for RL-for-LLMs is to develop *compute-allocation policies* that adapt the depth and halting of reasoning to instance difficulty and epistemic uncertainty. Current research has explored hard-coded reasoning levels in prompts [Agarwal et al., 2025a, Wen et al., 2025a, Zhu et al., 2025g], adaptive length-based reward shaping [Liu et al., 2025o, Yuan et al., 2025a], and the use of length penalties in the loss function [Aggarwal and Welleck, 2025, Xiang et al., 2025].

However, generalizing these approaches into a principled cost-performance trade-off remains an open question [Gan et al., 2025]. Teaching LRMs to be *resource-rational*, to reason longer only when the marginal utility justifies it, remains a central, unsolved problem for RL in language reasoning.

7.5. Teaching LLMs Latent Space Reasoning

CoT [Wei et al., 2022] encourages step-by-step reasoning by prompting models to articulate intermediate steps, improving both interpretability and accuracy. Recent research has combined CoT and RL to further improve reasoning quality, which samples long-form thought before answering for modeling training [Guo et al., 2025a]. However, current implementations often rely on token-level sampling [Cui et al., 2025a, Ouyang et al., 2022, Rafailov et al., 2023] in a discrete scalar space, which can act

as a bottleneck as the lost of meaningful semantic information in continuous space [Hua et al., 2024]. A recently proposed method, named Latent Space Reasoning (LSR) [Arriola et al., 2025, Geiping et al., 2025, Hao et al., 2024], may be more friendly for RL optimization. LSR operates reasoning in the continuous latent space of LLMs, facilitating more nuanced and fluid semantic reasoning. This characteristic contributes to smoother learning dynamics and a better integration with RL techniques. The combination of RL and LSR holds significant potential for the development of more powerful and adaptable reasoning models in the future. However, assessing the quality of continuous latent thought is more challenging than evaluating token-based thought. This will complicate the provision of accurate supervisory signals, such as rewards and advantages, which will become an open challenge against the combination of LSR and RL.

7.6. RL for LLMs Pre-training

Traditional pre-training relies on large text corpora and next-token prediction, and scaling this paradigm has already been shown to be central to the development of foundation models [Brown et al., 2020, Kaplan et al., 2020]. Emerging research now explores shifting RL earlier in the pipeline, applying it not only in post-training but also during pre-training itself. For instance, Reinforcement Pre-Training [Dong et al., 2025c] reconceptualizes next-token prediction as an RL problem with verifiable rewards derived from the corpus, reporting consistent gains that increase with available compute, thereby positioning RL as a promising scaling strategy for pre-training.

In parallel, open initiatives such as avataRL [tokenbender, 2025] demonstrate training language models from random initialization purely with RL, bootstrapping token-level rewards and employing iterative "referee" scoring, thus illustrating a concrete path toward RL-from-scratch training. It is consistent with the reincarnated RL paradigm [Agarwal et al., 2022], in which previously acquired computational knowledge (the pre-trained critic) is leveraged rather than training from the ground up. These developments sharpen a practical question: *how can RL-style pre-training be made cost-effective at scale?* Addressing this challenge will likely require reducing both the verifier burden and the costs associated with reward engineering, which appear to be critical for scaling RL-based pre-training. Moreover, this line of research is closely related to unsupervised reward design introduced in § 3.1.4, raising important questions about *how to obtain rewards that are both scalable and reliable*.

7.7. RL for Diffusion-based LLMs

Diffusion Large Language Models (DLLMs) [Cheng et al., 2025c, Labs et al., 2025, Nie et al., 2025, Tae et al., 2025, Xie et al., 2025f, Ye et al., 2025c] represent an emerging paradigm in language generation. Compared to autoregressive (AR) models, DLLMs offer advantages including superior decoding efficiency and a greater potential for self-correction through multiple rounds of diffusion. Initial efforts have begun to explore RL for DLLMs [Borso et al., 2025, Gong et al., 2025, Yang et al., 2025d], yet several key issues remain unresolved.

A central challenge in applying RL to DLLMs lies in accurately and efficiently estimating log probabilities of sampled responses. This is due to a fundamental difference in how autoregressive models and diffusion language models inherently model the likelihood of samples. AR models generate sequences through next-token prediction and factorize joint probabilities via the chain rule, enabling straightforward left-to-right sampling. However, DLLMs approximate likelihood optimization by maximizing the Evidence Lower Bound (ELBO). ELBO involves a double expectation over diffusion timesteps and masked data, and typically demands extensive sampling to achieve accurate estimates; otherwise, it introduces high variance during preference optimization. Although methods like the one-step estimator in [Zhao et al., 2025c] and the sampling allocation strategy in [Zhu et al., 2025b]

have been proposed to mitigate variance, efficient and accurate ELBO estimation remains an open problem for on-policy learning.

Furthermore, the existence of multiple feasible decoding trajectories in DLLMs introduces an additional research dimension: leveraging RL to guide the model toward optimal sampling traces. This requires designing effective reward functions for intermediate denoising steps. For example, He et al. [2025c] formulate denoising as a multi-step decision problem and applies reward models to intermediate states, [Wang et al., 2025o] proposed a diffusion-based value model that computes prefix-conditioned, token-wise advantages to enable trajectory-level rewards, while Song et al. [2025c] utilize edit-distance-based rewards to maximize decoding efficiency. Future work may also draw inspiration from RL techniques developed for continuous diffusion models in computer vision [Black et al., 2024b, Xue et al., 2025, Yang et al., 2024b], potentially paving the way toward a unified multimodal framework.

7.8. RL for LLMs in Scientific Discovery

Recent research has shown that involving RL can improve the performance of LLMs on reasoning-heavy scientific tasks, in some cases even allowing them to surpass specialized methods [Fallahpour et al., 2025, Fang et al., 2025c, Narayanan et al., 2025, Rizvi et al., 2025]. In domains such as biology and chemistry, a core challenge for RL is performing result verification at scale, a process conventionally dependent on wet lab experimentation. Several existing methods have focused on replacing or supplementing experimental verification: Pro-1 [Hla, 2025] uses a Rosetta energy function as the reward function for optimizing protein stability, and rbio1 [Istrate et al., 2025] verifies gene perturbation result predictions using biological models and external knowledge sources.

Much room for exploration remains for both reward formulation and improving the oracle models themselves. Related to this is the broader problem of constructing suitable RL environments that support rapid experimentation-feedback loops. Agentic systems such as Coscientist [Boiko et al., 2023] and Robin [Ghareeb et al., 2025] have gained success through lab-in-the-loop verification, but such sparse, delayed, and costly feedback signals are impractical for directly training the underlying LLM. *In silico* simulations of experimental environments, for instance perturbation response prediction at the cellular level [Bunne et al., 2024, Noutahi et al., 2025], represent a potential path forward. However, many of these systems are far from sufficient for replacing realistic lab environments due to their limited scope and critical lack of accuracy and generalizability [Ahlmann-Eltze et al., 2025, Kedzierska et al., 2023]. Other lines of research have explored incorporating domain-specific models into LLM training to handle scientific data [Fallahpour et al., 2025] and developing generalist models capable of a suite of well-defined tasks [Bigaud et al., 2025, Narayanan et al., 2025]. These directions, coupled with advances in general RL methodology, will continue expanding the use cases of LLMs from narrowly defined tasks to complex interactions with open-ended objectives, enabling them to more substantially contribute to novel discoveries.

7.9. RL for Architecture-Algorithm Co-Design

Most current RL pipelines for LLMs assume a dense Transformer [Vaswani et al., 2017] or Mixture-of-Experts (MoE) [Dai et al., 2024, Jiang et al., 2024, Shazeer et al., 2017] backbone, optimizing rewards that are almost exclusively tied to task accuracy. As a result, architectural degrees of freedom, and their hardware implications are left outside the learning loop. In parallel, a new wave of hardware, architecture co-design has emerged (e.g., hardware-aligned sparse attention as in DeepSeek's NSA [Yuan et al., 2025b] and model–system co-design in Step-3 [Wang et al., 2025a]), indicating that greater efficiency and capability can be achieved by aligning model structure with

computational substrates.

We argue that making architecture a first-class action space in RL represents an open and high-impact challenge for next-generation LLMs. For instance, reinforced MoE approaches could enable models to learn routing policies, expert activation, capacity allocation, or sparsity patterns during RL, optimizing not only for task reward, but also for hardware-aware objectives such as latency, memory traffic, energy consumption, and activation budgets. In this framing, RL is tasked with learning to "reason" not only over tokens [Guo et al., 2025a], but also across parameters and modules, dynamically adapting the model's topology to each prompt's difficulty and to real-time compute constraints. This perspective goes beyond classic RL-based neural architecture search (NAS) [Zoph and Le, 2016], which typically finds a fixed architecture for a given task or dataset. In contrast, reinforced MoE focuses on optimizing routing and modular adaptation per input during inference [Han et al., 2021], potentially yielding both greater efficiency and flexibility. Key open questions include designing robust multi-objective reward functions that avoid trivial solutions (e.g., all-expert sparsity), achieving stable credit assignment when architectural actions modify network topology, and amortizing architecture policy learning across prompts, tasks, and deployment scales. Addressing these challenges will be crucial for enabling truly integrated architecture–algorithm co-optimization in future LLMs.

8. Conclusion

We survey recent advances in RL for LRMs with a particular emphasis on reasoning, effectively transforming LLMs into LRMs. In contrast to prior approaches such as RLHF or DPO, which are primarily designed for human alignment, our focus is on RLVR for LLMs. RLVR enhances the reasoning abilities of LLMs by providing direct outcome-level rewards. Firstly, we present the core components of RLVR, including reward design, policy optimization, and sampling strategies. We summarize multiple research directions and existing work for each section. And then we discuss several of the most hotly debated issues in RL training for LLMs. In addition, we introduce training resources for RL of LLMs, covering static datasets, dynamic environments, and RL infrastructure. Finally, we review downstream applications of RL in LLMs across various scenarios and highlight several promising research directions aimed at achieving super-intelligence through RL-based LLMs.

Author Contributions

We present below the contributions of all participating authors, specifying the primary responsible individual and the contributing participants for each section.

- **Corresponding Author:** Biqing Qi, Ning Ding, Bowen Zhou

- **Project Lead:** Kaiyan Zhang, Yuxin Zuo

- **Introduction:** Kaiyan Zhang

- **Preliminaries:**

  - Background: Kaiyan Zhang
  - Frontier Models: Shang Qu, Yuru Wang (Survey)
  - Related Surveys: Yuxin Zuo

- **Foundational Components:**

  - Verifiable Rewards: Yuxin Zuo
  - Generative Rewards: Bingxiang He, Sihang Zeng (Draft)
  - Dense Rewards: Runze Liu, Yu Fu (Turn-level)
  - Unsupervised Rewards: Bingxiang He, Yuxin Zuo (Survey)
  - Reward Shaping: Kaiyan Zhang
  - Policy Gradient Objective: Youbang Sun, Kaiyan Zhang (Formula)
  - Critic-based Algorithms: Youbang Sun, Kaiyan Zhang (Formula)
  - Critic-Free Algorithms: Youbang Sun, Kaiyan Zhang (Formula)
  - Off-policy Optimization: Xingtai Lv, Yu Fu (Replay Buffer)
  - Regularization Objectives: Yuchen Zhang, Bingxiang He (Entropy), Yuxin Zuo (Survey)
  - Dynamic and Structured Sampling: Yuchen Fan, Xuekai Zhu (Efficiency-oriented Sampling)
  - Sampling Hyper-parameters: Yuxin Zuo, Bingxiang He (Survey)

- **Foundational Problems:** Kaiyan Zhang, Yuxin Zuo

- **Training Resource:**

  - Static Corpus: Kai Tian, Zhenzhao Yuan (Survey)
  - Dynamic Environment: Che Jiang
  - RL Infrastructure: Kaiyan Zhang

- **Applications:**

  - Coding Tasks: Pengfei Li, Xiang Xu (Tool-Integrated Reasoning)
  - Agentic Tasks: Yuchen Fan, Xinwei Long (GUI/Computer-use)
  - Multimodal Tasks: Guoli Jia, Fangfu Liu (Video/3D), Xinwei Long (Image)
  - Multi-Agent Systems: Shijie Wang
  - Robotics Tasks: Haozhan Li
  - Medical Tasks: Sihang Zeng, Jiaze Ma (Survey)

- **Future Directions:**

    - Che Jiang (CRL), Yu Fu (MemRL), Ermo Hua (Latent), Yuxin Zuo (Pre-Training), Yihao Liu (Diffusion), Shang Qu (Scientific), Kaiyan Zhang (Rest of All)

- **Other Contributions:**

    - Figures: Yuru Wang, Kaiyan Zhang, Yuxin Zuo
    - Review and Editing: Zhiyuan Ma, Ganqu Cui, Huayu Chen, Weize Chen, Yafu Li, Xiaoye Qu, Junqi Gao, Dong Li, Zonglin Li, and all above authors

References

a-m team. Am-deepseek-r1-0528-distilled, June 2025. URL https://github.com/a-m-team/a-m-models.

Marah Abdin, Sahaj Agarwal, Ahmed Awadallah, Vidhisha Balachandran, Harkirat Behl, Lingjiao Chen, Gustavo de Rosa, Suriya Gunasekar, Mojan Javaheripi, Neel Joshi, et al. Phi-4-reasoning technical report. *arXiv preprint arXiv:2504.21318*, 2025.

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Reincarnating reinforcement learning: Reusing prior computation to accelerate progress. *Advances in neural information processing systems*, 35:28955–28971, 2022.

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. gpt-oss-120b & gpt-oss-20b model card. *arxiv preprint arXiv: 2508.10925*, 2025a.

Shivam Agarwal, Zimin Zhang, Lifan Yuan, Jiawei Han, and Hao Peng. The unreasonable effectiveness of entropy minimization in llm reasoning. *arXiv preprint arXiv:2505.15134*, 2025b.

Pranjal Aggarwal and Sean Welleck. L1: Controlling how long a reasoning model thinks with reinforcement learning. *arXiv preprint arXiv:2503.04697*, 2025.

Armen Aghajanyan, Lili Yu, Alexis Conneau, Wei-Ning Hsu, Karen Hambardzumyan, Susan Zhang, Stephen Roller, Naman Goyal, Omer Levy, and Luke Zettlemoyer. Scaling laws for generative mixed-modal language models. In *International Conference on Machine Learning*, pages 265–279. PMLR, 2023.

Constantin Ahlmann-Eltze, Wolfgang Huber, and Simon Anders. Deep-learning-based gene perturbation effect prediction does not yet outperform simple linear baselines. *Nature Methods*, pages 1657–1661, 2025.

Wasi Uddin Ahmad, Sean Narenthiran, Somshubra Majumdar, Aleksander Ficek, Siddhartha Jain, Jocelyn Huang, Vahid Noroozi, and Boris Ginsburg. Opencodereasoning: Advancing data distillation for competitive coding. *arXiv preprint arXiv:2504.01943*, 2025.

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms. *arXiv preprint arXiv:2402.14740*, 2024.

Moonshot AI. Kimi-researcher: End-to-end rl training for emerging agentic capabilities. https://moonshotai.github.io/Kimi-Researcher/, 2025. Accessed: 2025-08-13.

Alon Albalak, Duy Phung, Nathan Lile, Rafael Rafailov, Kanishk Gandhi, Louis Castricato, Anikait Singh, Chase Blagden, Violet Xiang, Dakota Mahan, et al. Big-math: A large-scale, high-quality math dataset for reinforcement learning in language models. *arXiv preprint arXiv:2502.17387*, 2025.

Chenxin An, Zhihui Xie, Xiaonan Li, Lei Li, Jun Zhang, Shansan Gong, Ming Zhong, Jingjing Xu, Xipeng Qiu, Mingxuan Wang, and Lingpeng Kong. Polaris: A post-training recipe for scaling reinforcement learning on advanced reasoning models, 2025. URL https://hkunlp.github.io/blog/2025/Polaris.

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3674–3683, 2018.

Zachary Ankner, Mansheej Paul, Brandon Cui, Jonathan D Chang, and Prithviraj Ammanabrolu. Critique-out-loud reward models. *arXiv preprint arXiv:2408.11791*, 2024.

Anthropic. Claude 3.7 sonnet and claude code, 2025a. URL https://www.anthropic.com/news/claude-3-7-sonnet.

Anthropic. Claude opus 4.1, 2025b. URL https://www.anthropic.com/claude/opus.

Iván Arcuschin, Jett Janiak, Robert Krzyzanowski, Senthooran Rajamanoharan, Neel Nanda, and Arthur Conmy. Chain-of-thought reasoning in the wild is not always faithful. *arXiv preprint arXiv:2503.08679*, 2025.

Daman Arora and Andrea Zanette. Training language models to reason efficiently. *arXiv preprint arXiv:2502.04463*, 2025.

Pranav Arora, Rohan Gupta, and Kavya Patel. Gazal-r1: Scaling medical reasoning with grpo and multi-component reward design, 2025. URL https://arxiv.org/abs/2506.21594.

Marianne Arriola, Aaron Gokaslan, Justin T Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. *arXiv preprint arXiv:2503.09573*, 2025.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Ibragim Badertdinov, Alexander Golubev, Maksim Nekrashevich, Anton Shevtsov, Simon Karasik, Andrei Andriushchenko, Maria Trofimova, Daria Litvintseva, and Boris Yangel. Swe-rebench: An automated pipeline for task collection and decontaminated evaluation of software engineering agents. *arXiv preprint arXiv:2505.20411*, 2025.

Lei Bai, Zhongrui Cai, Maosong Cao, Weihan Cao, Chiyu Chen, Haojiong Chen, Kai Chen, Pengcheng Chen, Ying Chen, Yongkang Chen, et al. Intern-s1: A scientific multimodal foundation model. *arXiv preprint arXiv:2508.15763*, 2025.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022a.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022b.

Baidu-ERNIE-Team. Ernie 4.5 technical report. https://ernie.baidu.com/blog/publication/ERNIE_Technical_Report.pdf, 2025.

Bowen Baker, Joost Huizinga, Leo Gao, Zehao Dou, Melody Y Guan, Aleksander Madry, Wojciech Zaremba, Jakub Pachocki, and David Farhi. Monitoring reasoning models for misbehavior and the risks of promoting obfuscation. *arXiv preprint arXiv:2503.11926*, 2025.

Philip J. Ball, Jakob Bauer, Frank Belletti, Bethanie Brownfield, Ariel Ephrat, and et al. Genie 3: A new frontier for world models. https://deepmind.google/discover/blog/genie-3-a-new-frontier-for-world-models/, 2025.

David Bani-Harouni. Language agents for hypothesis-driven clinical decision making with reinforcement learning, 2025. URL https://arxiv.org/abs/2506.13474.

Abdelhakim Benechehab, Youssef Attia El Hili, Ambroise Odonnat, Oussama Zekri, Albert Thomas, Giuseppe Paolo, Maurizio Filippone, Ievgen Redko, and Balázs Kégl. Zero-shot model-based reinforcement learning using large language models. *arXiv preprint arXiv:2410.11711*, 2024.

Akhiad Bercovich, Itay Levy, Izik Golan, Mohammad Dabbah, Ran El-Yaniv, Omri Puny, Ido Galil, Zach Moshe, Tomer Ronen, Najeeb Nabwani, et al. Llama-nemotron: Efficient reasoning models. *arxiv preprint arXiv: 2505.00949*, 2025.

Glen Berseth, Zhiwei Zhang, Grace Zhang, Chelsea Finn, and Sergey Levine. Comps: Continual meta policy search. *arXiv preprint arXiv:2112.04467*, 2021.

Nathan Bigaud, Vincent Cabeli, Meltem Gürel, Arthur Pignet, John Klein, Gilles Wainrib, and Eric Durand. OwkinZero: Accelerating biological discovery with AI. *arXiv preprint arXiv: 2508.16315*, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. pi0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024a.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024b. URL https://openreview.net/forum?id=YCWjhGrJFD.

Daniil A Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578, 2023.

Umberto Borso, Davide Paglieri, Jude Wells, and Tim Rocktäschel. Preference-based alignment of discrete diffusion models. *arXiv preprint arXiv:2503.08295*, 2025.

Michael Bowling, John D Martin, David Abel, and Will Dabney. Settling the reward hypothesis. In *International Conference on Machine Learning*, pages 3003–3020. PMLR, 2023.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

Charlotte Bunne, Yusuf Roohani, Yanay Rosen, Ankit Gupta, Xikun Zhang, Marcel Roed, Theo Alexandrov, Mohammed AlQuraishi, Patricia Brennan, Daniel B Burkhardt, et al. How to build the virtual cell with artificial intelligence: Priorities and opportunities. *Cell*, 187(25):7045–7063, 2024.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, et al. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. *arXiv preprint arXiv:2312.09390*, 2023.

Lucian Busoniu, Robert Babuska, and Bart De Schutter. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(2):156–172, 2008.

Lorenzo Canese, Gian Carlo Cardarilli, Luca Di Nunzio, Rocco Fazzolari, Daniele Giardino, Marco Re, and Sergio Spanò. Multi-agent reinforcement learning: A review of challenges and applications. *Applied Sciences*, 11(11):4948, 2021.

Meng Cao, Haoze Zhao, Can Zhang, Xiaojun Chang, Ian Reid, and Xiaodan Liang. Ground-r1: Incentivizing grounded visual reasoning via reinforcement learning. *arXiv preprint arXiv:2505.20272*, 2025.

Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal Patwardhan, et al. Mle-bench: Evaluating machine learning agents on machine learning engineering. *arXiv preprint arXiv:2410.07095*, 2024.

Olivier Chapelle and Alexander Zien. Semi-supervised classification by low density separation. In *International workshop on artificial intelligence and statistics*, pages 57–64. PMLR, 2005.

Aili Chen, Aonian Li, Bangwei Gong, Binyang Jiang, Bo Fei, Bo Yang, Boji Shan, Changqing Yu, Chao Wang, Cheng Zhu, et al. Minimax-m1: Scaling test-time compute efficiently with lightning attention. *arXiv preprint arXiv:2506.13585*, 2025a.

Ding Chen, Qingchen Yu, Pengyuan Wang, Wentao Zhang, Bo Tang, Feiyu Xiong, Xinchi Li, Minchuan Yang, and Zhiyu Li. xverify: Efficient answer verifier for reasoning model evaluations. *arXiv preprint arXiv:2504.10481*, 2025b.

Guangyao Chen, Siwei Dong, Yu Shu, Ge Zhang, Jaward Sesay, Börje F Karlsson, Jie Fu, and Yemin Shi. Autoagents: A framework for automatic agent generation. *arXiv preprint arXiv:2309.17288*, 2023a.

Huayu Chen, Kaiwen Zheng, Qinsheng Zhang, Ganqu Cui, Yin Cui, Haotian Ye, Tsung-Yi Lin, Ming-Yu Liu, Jun Zhu, and Haoxiang Wang. Bridging supervised learning and reinforcement learning in math reasoning. *arXiv preprint arXiv:2505.18116*, 2025c.

Jiangjie Chen, Qianyu He, Siyu Yuan, Aili Chen, Zhicheng Cai, Weinan Dai, Hongli Yu, Qiying Yu, Xuefeng Li, Jiaze Chen, et al. Enigmata: Scaling logical reasoning in large language models with synthetic verifiable puzzles. *arXiv preprint arXiv:2505.19914*, 2025d.

Junying Chen, Zhenyang Cai, Ke Ji, Xidong Wang, Wanlong Liu, Rongsheng Wang, Jianye Hou, and Benyou Wang. Huatuogpt-o1, towards medical complex reasoning with llms, 2024a. URL https://arxiv.org/abs/2412.18925.

Lei Chen, Xuanle Zhao, Zhixiong Zeng, Jing Huang, Liming Zheng, Yufeng Zhong, and Lin Ma. Breaking the sft plateau: Multimodal structured reinforcement learning for chart-to-code generation. *arXiv preprint arXiv:2508.13587*, 2025e.

Lei Chen, Xuanle Zhao, Zhixiong Zeng, Jing Huang, Yufeng Zhong, and Lin Ma. Chart-r1: Chain-of-thought supervision and reinforcement for advanced chart reasoner. *arXiv preprint arXiv:2507.15509*, 2025f.

Liang Chen, Hongcheng Gao, Tianyu Liu, Zhiqi Huang, Flood Sung, Xinyu Zhou, Yuxin Wu, and Baobao Chang. G1: Bootstrapping perception and reasoning abilities of vision-language model via reinforcement learning. *arXiv preprint arXiv:2505.13426*, 2025g.

Liang Chen, Xueting Han, Li Shen, Jing Bai, and Kam-Fai Wong. Beyond two-stage training: Cooperative sft and rl for llm reasoning, 2025h. URL https://arxiv.org/abs/2509.06948.

Lili Chen, Mihir Prabhudesai, Katerina Fragkiadaki, Hao Liu, and Deepak Pathak. Self-questioning language models. *arXiv preprint arXiv:2508.03682*, 2025i.

Minghan Chen, Guikun Chen, Wenguan Wang, and Yi Yang. Seed-grpo: Semantic entropy enhanced grpo for uncertainty-aware policy optimization. *arXiv preprint arXiv:2505.12346*, 2025j.

Mingyang Chen, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Haofen Wang, Jeff Z Pan, Wen Zhang, Huajun Chen, Fan Yang, et al. Learning to reason with search for llms via reinforcement learning. *arXiv preprint arXiv:2503.19470*, 2025k.

Nuo Chen, Zhiyuan Hu, Qingyun Zou, Jiaying Wu, Qian Wang, Bryan Hooi, and Bingsheng He. Judgelrm: Large reasoning models as a judge. *arXiv preprint arXiv:2504.00050*, 2025l.

Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *arXiv preprint arXiv:2503.09567*, 2025m.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Qiwei Liang, Zixuan Li, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025n.

Xiaoyin Chen, Jiarui Lu, Minsu Kim, Dinghuai Zhang, Jian Tang, Alexandre Piché, Nicolas Gontier, Yoshua Bengio, and Ehsan Kamalloo. Self-evolving curriculum for llm reasoning. *arXiv preprint arXiv:2505.14970*, 2025o.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, et al. Do not think that much for 2+ 3=? on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187*, 2024b.

Xiusi Chen, Gaotang Li, Ziqi Wang, Bowen Jin, Cheng Qian, Yu Wang, Hongru Wang, Yu Zhang, Denghui Zhang, Tong Zhang, et al. Rm-r1: Reward modeling as reasoning. *arXiv preprint arXiv:2505.02387*, 2025p.

Yang Chen, Zhuolin Yang, Zihan Liu, Chankyu Lee, Peng Xu, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. Acereason-nemotron: Advancing math and code reasoning through reinforcement learning. *arXiv preprint arXiv:2505.16400*, 2025q.

Yongchao Chen, Yueying Liu, Junwei Zhou, Yilun Hao, Jingquan Wang, Yang Zhang, and Chuchu Fan. R1-code-interpreter: Training llms to reason with code via supervised and reinforcement learning. *arXiv preprint arXiv:2505.21668*, 2025r.

Yuanpei Chen, Yiran Geng, Fangwei Zhong, Jiaming Ji, Jiechuang Jiang, Zongqing Lu, Hao Dong, and Yaodong Yang. Bi-dexhands: Towards human-level bimanual dexterous manipulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5):2804–2818, 2023b.

Yuhui Chen, Shuai Tian, Shugao Liu, Yingting Zhou, Haoran Li, and Dongbin Zhao. Conrft: A reinforced fine-tuning method for vla models via consistency policy. *arXiv preprint arXiv:2502.05450*, 2025s.

Yukang Chen, Wei Huang, Baifeng Shi, Qinghao Hu, Hanrong Ye, Ligeng Zhu, Zhijian Liu, Pavlo Molchanov, Jan Kautz, Xiaojuan Qi, et al. Scaling rl to long videos. *arXiv preprint arXiv:2507.07966*, 2025t.

Yuyang Chen, Kaiyan Zhao, Yiming Wang, Ming Yang, Jian Zhang, and Xiaoguang Niu. Enhancing llm agents for code generation with possibility and pass-rate prioritized experience replay. *arXiv preprint arXiv:2410.12236*, 2024c.

Zhipeng Chen, Yingqian Min, Beichen Zhang, Jie Chen, Jinhao Jiang, Daixuan Cheng, Wayne Xin Zhao, Zheng Liu, Xu Miao, Yang Lu, et al. An empirical study on eliciting and improving r1-like reasoning models. *arXiv preprint arXiv:2503.04548*, 2025u.

Zhipeng Chen, Xiaobo Qin, Youbin Wu, Yue Ling, Qinghao Ye, Wayne Xin Zhao, and Guang Shi. Pass@k training for adaptively balancing exploration and exploitation of large reasoning models. *arXiv preprint arXiv:2508.10751*, 2025v.

Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Wayne Xin Zhao, Zhenliang Zhang, and Furu Wei. Reasoning with exploration: An entropy perspective. *arXiv preprint arXiv:2506.14758*, 2025a.

Jie Cheng, Ruixi Qiao, Lijun Li, Chao Guo, Junle Wang, Gang Xiong, Yisheng Lv, and Fei-Yue Wang. Stop summation: Min-form credit assignment is all process reward model needs for reasoning. *arXiv preprint arXiv:2504.15275*, 2025b.

Shuang Cheng, Yihan Bian, Dawei Liu, Yuhua Jiang, Yihao Liu, Linfeng Zhang, Wenghai Wang, Qipeng Guo, Kai Chen, Biqing Qi*, and Bowen Zhou. Sdar: A synergistic diffusion–autoregression paradigm for scalable sequence generation, 2025c. URL https://github.com/JetAstra/SDAR.

Zhoujun Cheng, Shibo Hao, Tianyang Liu, Fan Zhou, Yutao Xie, Feng Yao, Yuexin Bian, Yonghao Zhuang, Nilabjo Dey, Yuheng Zha, et al. Revisiting reinforcement learning for llm reasoning from a cross-domain perspective. *arXiv preprint arXiv:2506.14965*, 2025d.

Maxime Chevalier-Boisvert, Bolun Dai, Mark Towers, Rodrigo Perez-Vicente, Lucas Willems, Salem Lahlou, Suman Pal, Pablo Samuel Castro, and Jordan Terry. Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks. *Advances in Neural Information Processing Systems*, 36:73383–73394, 2023.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready ai agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*, 2025.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. *arXiv preprint arXiv:2501.17161*, 2025a.

Xu Chu, Xinrong Chen, Guanyu Wang, Zhijie Tan, Kui Huang, Wenyu Lv, Tong Mo, and Weiping Li. Qwen look again: Guiding vision-language reasoning models to re-attention visual information. *arXiv preprint arXiv:2505.23558*, 2025b.

Jiwan Chung, Junhyeok Kim, Siyeol Kim, Jaeyoung Lee, Min Soo Kim, and Youngjae Yu. Don't look only once: Towards multimodal interactive reasoning with selective visual revisitation. *arXiv preprint arXiv:2505.18842*, 2025.

Taco Cohen, David W Zhang, Kunhao Zheng, Yunhao Tang, Remi Munos, and Gabriel Synnaeve. Soft policy optimization: Online off-policy rl for sequence models. *arXiv preprint arXiv:2503.05453*, 2025.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arxiv preprint arXiv: 2507.06261*, 2025.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, et al. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025a.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, et al. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025b.

Damai Dai, Chengqi Deng, Chenggang Zhao, RX Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Yu Wu, et al. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. *arXiv preprint arXiv:2401.06066*, 2024.

Muzhi Dai, Chenxu Yang, and Qingyi Si. S-grpo: Early exit via reinforcement learning in reasoning models. *arXiv preprint arXiv:2505.07686*, 2025a.

Yaxun Dai, Wenxuan Xie, Xialie Zhuang, Tianyu Yang, Yiying Yang, Haiqin Yang, Yuhang Zhao, Pingfu Chao, and Wenhao Jiang. Reex-sql: Reasoning with execution-aware reinforcement learning for text-to-sql. *arXiv preprint arXiv:2505.12768*, 2025b.

Jisheng Dang, Jingze Wu, Teng Wang, Xuanhui Lin, Nannan Zhu, Hongbo Chen, Wei-Shi Zheng, Meng Wang, and Tat-Seng Chua. Reinforcing video reasoning with focused thinking. *arXiv preprint arXiv:2505.24718*, 2025.

Alan Dao and Thinh Le. Rezero: Enhancing llm search ability by trying one-more-time. *arXiv preprint arXiv:2504.11001*, 2025.

Alan Dao and Dinh Bach Vu. Jan-nano technical report. *arXiv preprint arXiv:2506.22760*, 2025.

Antoine Dedieu, Joseph Ortiz, Xinghua Lou, Carter Wendelken, Wolfgang Lehrach, J Swaroop Guntupalli, Miguel Lazaro-Gredilla, and Kevin Patrick Murphy. Improving transformer world models for data-efficient rl. In *ICLR 2025 Workshop on World Models: Understanding, Modelling and Scaling*, 2025.

Jia Deng, Jie Chen, Zhipeng Chen, Daixuan Cheng, Fei Bai, Beichen Zhang, Yinqian Min, Yanzipeng Gao, Wayne Xin Zhao, and Ji-Rong Wen. From trial-and-error to improvement: A systematic analysis of llm exploration mechanisms in rlvr. *arXiv preprint arXiv:2508.07534*, 2025a.

Yong Deng, Guoqing Wang, Zhenzhe Ying, Xiaofeng Wu, Jinzhen Lin, Wenwen Xiong, Yuqin Dai, Shuo Yang, Zhanwei Zhang, Qiwen Wang, et al. Atom-searcher: Enhancing agentic deep research via fine-grained atomic thought reward. *arXiv preprint arXiv:2508.12800*, 2025b.

Hongxin Ding, Baixiang Huang, and Yue Fang. Promed: Shapley information gain guided reinforcement learning for proactive medical llms, 2025. URL https://arxiv.org/abs/2508.13514.

Ziluo Ding, Wanpeng Zhang, Junpeng Yue, Xiangjun Wang, Tiejun Huang, and Zongqing Lu. Entity divider with language grounding in multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 8103–8119. PMLR, 2023.

Dai Do, Manh Nguyen, Svetha Venkatesh, and Hung Le. Sparft: Self-paced reinforcement fine-tuning for large language models. *arXiv preprint arXiv:2508.05015*, 2025.

Shibhansh Dohare, J Fernando Hernandez-Garcia, Qingfeng Lan, Parash Rahman, A Rupam Mahmood, and Richard S Sutton. Loss of plasticity in deep continual learning. *Nature*, 632(8026):768–774, 2024.

Guanting Dong, Yifei Chen, Xiaoxi Li, Jiajie Jin, Hongjin Qian, Yutao Zhu, Hangyu Mao, Guorui Zhou, Zhicheng Dou, and Ji-Rong Wen. Tool-star: Empowering llm-brained multi-tool reasoner via reinforcement learning. *arXiv preprint arXiv:2505.16410*, 2025a.

Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, et al. Agentic reinforced policy optimization. *arXiv preprint arXiv:2507.19849*, 2025b.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. Raft: Reward ranked finetuning for generative foundation model alignment. *arXiv preprint arXiv:2304.06767*, 2023.

Qingxiu Dong, Li Dong, Yao Tang, Tianzhu Ye, Yutao Sun, Zhifang Sui, and Furu Wei. Reinforcement pre-training. *arXiv preprint arXiv:2506.08007*, 2025c.

Ali Dorri, Salil S Kanhere, and Raja Jurdak. Multi-agent systems: A survey. *Ieee Access*, 6:28573–28593, 2018.

Shihan Dou, Muling Wu, Jingwen Xu, Rui Zheng, Tao Gui, Qi Zhang, and Xuanjing Huang. Improving rl exploration for llm reasoning through retrospective replay. *arXiv preprint arXiv:2504.14363*, 2025.

Mingzhe Du, Luu Anh Tuan, Yue Liu, Yuhao Qing, Dong Huang, Xinyi He, Qian Liu, Zejun Ma, and See-kiong Ng. Afterburner: Reinforcement learning facilitates self-improving code efficiency optimization. *arXiv preprint arXiv:2505.23387*, 2025a.

Yong Du, Yuchen Yan, Fei Tang, Zhengxi Lu, Chang Zong, Weiming Lu, Shengpei Jiang, and Yongliang Shen. Test-time reinforcement learning for gui grounding via region consistency. *arXiv preprint arXiv:2508.05615*, 2025b.

Chengqi Duan, Rongyao Fang, Yuqing Wang, Kun Wang, Linjiang Huang, Xingyu Zeng, Hongsheng Li, and Xihui Liu. Got-r1: Unleashing reasoning capability of mllm for visual generation with reinforcement learning. *arXiv preprint arXiv:2505.17022*, 2025.

Sam Earle, Graham Todd, Yuchen Li, Ahmed Khalifa, Muhammad Umair Nasir, Zehua Jiang, Andrzej Banburski-Fahey, and Julian Togelius. Puzzlejax: A benchmark for reasoning and learning, 2025. URL https://arxiv.org/abs/2508.16821.

Ahmed El-Kishky, Alexander Wei, Andre Saraiva, Borys Minaiev, Daniel Selsam, David Dohan, Francis Song, Hunter Lightman, Ignasi Clavera, Jakub Pachocki, et al. Competitive programming with large reasoning models. *arXiv preprint arXiv:2502.06807*, 2025.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Benjamin Eysenbach and Sergey Levine. Maximum entropy rl (provably) solves some robust rl problems. *arXiv preprint arXiv:2103.06257*, 2021.

Hugging Face. Open r1: A fully open reproduction of deepseek-r1, january 2025. *URL https://github. com/huggingface/open-r1*, page 9, 2025.

Adibvafa Fallahpour, Andrew Magnuson, Purav Gupta, Shihao Ma, Jack Naimer, Arnav Shah, Haonan Duan, Omar Ibrahim, Hani Goodarzi, Chris J. Maddison, and Bo Wang. BioReason: Incentivizing multimodal biological reasoning within a DNA-LLM model. *arXiv preprint arXiv: 2505.23579*, 2025.

Run-Ze Fan, Zengzhi Wang, and Pengfei Liu. Megascience: Pushing the frontiers of post-training datasets for science reasoning. *arXiv preprint arXiv:2507.16812*, 2025a.

Tiantian Fan, Lingjun Liu, Yu Yue, Jiaze Chen, Chengyi Wang, Qiying Yu, Chi Zhang, Zhiqi Lin, Ruofei Zhu, Yufeng Yuan, et al. Truncated proximal policy optimization. *arXiv preprint arXiv:2506.15050*, 2025b.

Yuchen Fan, Kaiyan Zhang, Heng Zhou, Yuxin Zuo, Yanxu Chen, Yu Fu, Xinwei Long, Xuekai Zhu, Che Jiang, Yuchen Zhang, et al. Ssrl: Self-search reinforcement learning. *arXiv preprint arXiv:2508.10874*, 2025c.

Yue Fan, Xuehai He, Diji Yang, Kaizhi Zheng, Ching-Chen Kuo, Yuting Zheng, Sravana Jyothi Narayanaraju, Xinze Guan, and Xin Eric Wang. Grit: Teaching mllms to think with images. *arXiv preprint arXiv:2505.15879*, 2025d.

Ziqing Fan, Cheng Liang, Chaoyi Wu, Ya Zhang, Yanfeng Wang, and Weidi Xie. Chestx-reasoner: Advancing radiology foundation models with reasoning through step-by-step verification. *arXiv preprint arXiv:2504.20930*, 2025e.

Wenkai Fang, Shunyu Liu, Yang Zhou, Kongcheng Zhang, Tongya Zheng, Kaixuan Chen, Mingli Song, and Dacheng Tao. Serl: Self-play reinforcement learning for large language models with limited data. *arXiv preprint arXiv:2505.20347*, 2025a.

Xueji Fang, Liyuan Ma, Zhiyang Chen, Mingyuan Zhou, and Guo-jun Qi. Inflvg: Reinforce inference-time consistent long video generation with grpo. *arXiv preprint arXiv:2505.17574*, 2025b.

Yin Fang, Qiao Jin, Guangzhi Xiong, Bowen Jin, Xianrui Zhong, Siru Ouyang, Aidong Zhang, Jiawei Han, and Zhiyong Lu. Cell-o1: Training LLMs to solve single-cell reasoning puzzles with reinforcement learning, 2025c.

Wu Fei, Hao Kong, Shuxian Liang, Yang Lin, Yibo Yang, Jing Tang, Lei Chen, and Xiansheng Hua. Self-guided process reward optimization with redefined step-wise advantage for process reinforcement learning. *arXiv preprint arXiv:2507.01551*, 2025a.

Xiang Fei, Siqi Wang, Shu Wei, Yuxiang Nie, Wei Shi, Hao Feng, and Can Huang. Post-completion learning for language models. *arXiv preprint arXiv:2507.20252*, 2025b.

Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjun Zhong. Retool: Reinforcement learning for strategic tool use in llms. *arXiv preprint arXiv:2504.11536*, 2025a.

Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue. Video-r1: Reinforcing video reasoning in mllms. *arXiv preprint arXiv:2503.21776*, 2025b.

Sicheng Feng, Gongfan Fang, Xinyin Ma, and Xinchao Wang. Efficient reasoning models: A survey. *arXiv preprint arXiv:2504.10903*, 2025c.

Roya Firoozi, Johnathan Tucker, Stephen Tian, Anirudha Majumdar, Jiankai Sun, Weiyu Liu, Yuke Zhu, Shuran Song, Ashish Kapoor, Karol Hausman, et al. Foundation models in robotics: Applications, challenges, and the future. *The International Journal of Robotics Research*, 44(5):701–739, 2025.

Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Tingchen Fu, Jiawei Gu, Yafu Li, Xiaoye Qu, and Yu Cheng. Scaling reasoning, losing control: Evaluating instruction following in large reasoning models. *arXiv preprint arXiv:2505.14810*, 2025a.

Wei Fu, Jiaxuan Gao, Xujie Shen, Chen Zhu, Zhiyu Mei, Chuyi He, Shusheng Xu, Guo Wei, Jun Mei, Jiashu Wang, et al. Areal: A large-scale asynchronous reinforcement learning system for language reasoning. *arXiv preprint arXiv:2505.24298*, 2025b.

Yuqian Fu, Tinghong Chen, Jiajun Chai, Xihuai Wang, Songjun Tu, Guojun Yin, Wei Lin, Qichao Zhang, Yuanheng Zhu, and Dongbin Zhao. Srft: A single-stage method with supervised and reinforcement fine-tuning for reasoning. *arXiv preprint arXiv:2506.19767*, 2025c.

Yasuhiro Fujita. Experience replay with random reshuffling. *arXiv preprint arXiv:2503.02269*, 2025.

Marcos Fuster-Pena, David de Fitero-Dominguez, Antonio Garcia-Cabot, and Eva Garcia-Lopez. Repaca: Leveraging reasoning large language models for static automated patch correctness assessment. *arXiv preprint arXiv:2507.22580*, 2025.

Kushal Gajjar, Harshit Sikchi, Arpit Singh Gautam, Marc Hammons, and Saurabh Jha. Cognisql-r1-zero: Lightweight reinforced reasoning for efficient sql generation. *arXiv preprint arXiv:2507.06013*, 2025.

Zeyu Gan, Hao Yi, and Yong Liu. Cot-space: A theoretical framework for internal slow-thinking via reinforcement learning, 2025. URL https://arxiv.org/abs/2509.04027.

Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D Goodman. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars. *arXiv preprint arXiv:2503.01307*, 2025.

Jiaxuan Gao, Wei Fu, Minyang Xie, Shusheng Xu, Chuyi He, Zhiyu Mei, Banghua Zhu, and Yi Wu. Beyond ten turns: Unlocking long-horizon agentic search with large-scale asynchronous rl. *arXiv preprint arXiv:2508.07976*, 2025a.

Longxi Gao, Li Zhang, and Mengwei Xu. Uishift: Enhancing vlm-based gui agents through self-supervised reinforcement learning. *arXiv preprint arXiv:2505.12493*, 2025b.

Francisco Garcia and Philip S Thomas. A meta-mdp approach to exploration for lifelong reinforcement learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Jean-Baptiste Gaya, Thang Doan, Lucas Caccia, Laure Soulier, Ludovic Denoyer, and Roberta Raileanu. Building a subspace of policies for scalable continual learning. *arXiv preprint arXiv:2211.10445*, 2022.

Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian R Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, and Tom Goldstein. Scaling up test-time compute with latent reasoning: A recurrent depth approach. *arXiv preprint arXiv:2502.05171*, 2025.

Xinyu Geng, Peng Xia, Zhen Zhang, Xinyu Wang, Qiuchen Wang, Ruixue Ding, Chenxi Wang, Jialong Wu, Yida Zhao, Kuan Li, et al. Webwatcher: Breaking new frontiers of vision-language deep research agent. *arXiv preprint arXiv:2508.05748*, 2025.

Ali Essam Ghareeb, Benjamin Chang, Ludovico Mitchener, Angela Yiu, Caralyn J. Szostkiewicz, Jon M. Laurent, Muhammed T. Razzak, Andrew D. White, Michaela M. Hinks, and Samuel G. Rodriques. Robin: A multi-agent system for automating scientific discovery, 2025.

Majid Ghasemi, Amir Hossein Moosavi, and Dariush Ebrahimi. A comprehensive survey of reinforcement learning: From algorithms to practical challenges. *arXiv preprint arXiv:2411.18892*, 2024.

Elliot Glazer, Ege Erdil, Tamay Besiroglu, Diego Chicharro, Evan Chen, Alex Gunning, Caroline Falkman Olsson, Jean-Stanislas Denain, Anson Ho, Emily de Oliveira Santos, et al. Frontiermath: A benchmark for evaluating advanced mathematical reasoning in ai. *arXiv preprint arXiv:2411.04872*, 2024.

Shansan Gong, Ruixiang Zhang, Huangjie Zheng, Jiatao Gu, Navdeep Jaitly, Lingpeng Kong, and Yizhe Zhang. Diffucoder: Understanding and improving masked diffusion models for code generation. *arXiv preprint arXiv:2506.20639*, 2025.

Prasoon Goyal, Scott Niekum, and Raymond J Mooney. Using natural language for reward shaping in reinforcement learning. *arXiv preprint arXiv:1903.02020*, 2019.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Jihao Gu, Qihang Ai, Yingyao Wang, Pi Bu, Jingxuan Xing, Zekun Zhu, Wei Jiang, Ziming Wang, Yingxiu Zhao, Ming-Liang Zhang, et al. Mobile-r1: Towards interactive reinforcement learning for vlm-based mobile agent via task-level rewards. *arXiv preprint arXiv:2506.20332*, 2025.

Yu Gu, Kai Zhang, Yuting Ning, Boyuan Zheng, Boyu Gou, Tianci Xue, Cheng Chang, Sanjari Srivastava, Yanan Xie, Peng Qi, et al. Is your llm secretly a world model of the internet? model-based planning for web agents. *arXiv preprint arXiv:2411.06559*, 2024.

Zhong Guan, Likang Wu, Hongke Zhao, Jiahui Wang, and Le Wu. Recall-extend dynamics: Enhancing small language models through controlled exploration and refined offline integration. *arXiv preprint arXiv:2508.16677*, 2025.

Leon Guertler, Bobby Cheng, Simon Yu, Bo Liu, Leshem Choshen, and Cheston Tan. Textarena, 2025. URL https://arxiv.org/abs/2504.11442.

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Bing Liu, and Sean Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains. *arXiv preprint arXiv:2507.17746*, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shi-rong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025a.

Jiaxin Guo, Zewen Chi, Li Dong, Qingxiu Dong, Xun Wu, Shaohan Huang, and Furu Wei. Reward reasoning model. *arXiv preprint arXiv:2505.14674*, 2025b.

Yiran Guo, Lijie Xu, Jie Liu, Dan Ye, and Shuang Qiu. Segment policy optimization: Effective segment-level credit assignment in rl for large language models. *arXiv preprint arXiv:2505.23564*, 2025c.

Yongxin Guo, Wenbo Deng, Zhenglin Cheng, and Xiaoying Tang. G2 rpo-a: Guided group relative policy optimization with adaptive guidance. *arXiv preprint arXiv:2508.13023*, 2025d.

Zhihan Guo, Jiele Wu, Wenqian Cui, Yifei Zhang, Minda Hu, Yufei Wang, and Irwin King. From general to targeted rewards: Surpassing gpt-4 in open-ended long-context generation. *arXiv preprint arXiv:2506.16024*, 2025e.

Ziyu Guo, Renrui Zhang, Chengzhuo Tong, Zhizheng Zhao, Peng Gao, Hongsheng Li, and Pheng-Ann Heng. Can we generate images with cot? let's verify and reinforce image generation step by step. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025f.

Abhishek Gupta, Aldo Pacchiano, Yuexiang Zhai, Sham Kakade, and Sergey Levine. Unpacking reward shaping: Understanding the benefits of reward engineering on sample complexity. *Advances in Neural Information Processing Systems*, 35:15281–15295, 2022.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. Pmlr, 2018.

Dylan Hadfield-Menell, Smitha Milli, Pieter Abbeel, Stuart J Russell, and Anca Dragan. Inverse reward design. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/32fdab6559cdfa4f167f8c31b9199643-Paper.pdf.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Ai Han, Junxing Hu, Pu Wei, Zhiqian Zhang, Yuhang Guo, Jiawei Lu, and Zicheng Zhang. Joyagents-r1: Joint evolution dynamics for versatile multi-llm agents with reinforcement learning. *arXiv preprint arXiv:2506.19846*, 2025.

Yizeng Han, Gao Huang, Shiji Song, Le Yang, Honghui Wang, and Yulin Wang. Dynamic neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(11):7436–7456, 2021.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*, 2024.

Bingxiang He, Wenbin Zhang, Jiaxi Song, Cheng Qian, Zixuan Fu, Bowen Sun, Ning Ding, Haiwen Hong, Longtao Huang, Hui Xue, et al. Air: A systematic analysis of annotations, instructions, and response pairs in preference dataset. *arXiv preprint arXiv:2504.03612*, 2025a.

Feng He, Zijun Chen, Xinnian Liang, Tingting Ma, Yunqi Qiu, Shuangzhi Wu, and Junchi Yan. Protoreasoning: Prototypes as the foundation for generalizable reasoning in llms. *arXiv preprint arXiv:2506.15211*, 2025b.

Haoyu He, Katrin Renz, Yong Cao, and Andreas Geiger. Mdpo: Overcoming the training-inference divide of masked diffusion language models. *arXiv preprint arXiv:2508.13148*, 2025c.

Jujie He, Jiacai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, et al. Skywork open reasoner 1 technical report. *arXiv preprint arXiv:2505.22312*, 2025d.

Qianyu He, Siyu Yuan, Xuefeng Li, Mingxuan Wang, and Jiangjie Chen. Thinkdial: An open recipe for controlling reasoning effort in large language models. *arXiv preprint arXiv:2508.18773*, 2025e.

Tao He, Rongchuan Mu, Lizi Liao, Yixin Cao, Ming Liu, and Bing Qin. Good learners think their thinking: Generative prm makes large reasoning model more efficient math learner. *arXiv preprint arXiv:2507.23317*, 2025f.

Xiaoxuan He, Siming Fu, Yuke Zhao, Wanli Li, Jian Yang, Dacheng Yin, Fengyun Rao, and Bo Zhang. Tempflow-grpo: When timing matters for grpo in flow models. *arXiv preprint arXiv:2508.04324*, 2025g.

Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, et al. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. *arXiv preprint arXiv:2504.11456*, 2025h.

Michael Hla. Pro-1, 2025. URL https://michaelhla.com/blog/pro1.html.

Haitao Hong, Yuchen Yan, Xingyu Wu, Guiyang Hou, Wenqi Zhang, Weiming Lu, Yongliang Shen, and Jun Xiao. Cooper: Co-optimizing policy and reward models in reinforcement learning for large language models. *arXiv preprint arXiv:2508.05613*, 2025a.

Ilgee Hong, Changlong Yu, Liang Qiu, Weixiang Yan, Zhenghao Xu, Haoming Jiang, Qingru Zhang, Qin Lu, Xin Liu, Chao Zhang, et al. Think-rm: Enabling long-horizon reasoning in generative reward models. *arXiv preprint arXiv:2505.16265*, 2025b.

Jixiang Hong, Yiran Zhang, Guanzhong Wang, Yi Liu, Ji-Rong Wen, and Rui Yan. Reinforcing multimodal understanding and generation with dual self-rewards. *arXiv preprint arXiv:2506.07963*, 2025c.

Zhenyu Hou, Ziniu Hu, Yujiang Li, Rui Lu, Jie Tang, and Yuxiao Dong. Treerl: Llm reinforcement learning with on-policy tree search. *arXiv preprint arXiv:2506.11902*, 2025.

Haichuan Hu, Xiaochen Xie, and Quanjun Zhang. Repair-r1: Better test before repair. *arXiv preprint arXiv:2507.22853*, 2025a.

Jian Hu. Reinforce++: A simple and efficient approach for aligning large language models. *arXiv preprint arXiv:2501.03262*, 2025.

Jian Hu, Xibin Wu, Zilin Zhu, Weixun Wang, Dehao Zhang, Yu Cao, et al. Openrlhf: An easy-to-use, scalable and high-performance rlhf framework. *arXiv preprint arXiv:2405.11143*, 2024a.

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *arXiv preprint arXiv:2503.24290*, 2025b.

Lanxiang Hu, Mingjia Huo, Yuxuan Zhang, Haoyang Yu, Eric P Xing, Ion Stoica, Tajana Rosing, Haojian Jin, and Hao Zhang. lmgame-bench: How good are llms at playing games? *arXiv preprint arXiv:2505.15146*, 2025c.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, et al. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*, 2024b.

Yujing Hu, Weixun Wang, Hangtian Jia, Yixiang Wang, Yingfeng Chen, Jianye Hao, Feng Wu, and Changjie Fan. Learning to utilize shaping rewards: A new approach of reward shaping. *Advances in Neural Information Processing Systems*, 33:15931–15941, 2020.

Zhiting Hu and Tianmin Shu. Language models, agent models, and world models: The law for machine reasoning and planning. *arXiv preprint arXiv:2312.05230*, 2023.

Ermo Hua, Biqing Qi, Kaiyan Zhang, Yue Yu, Ning Ding, Xingtai Lv, Kai Tian, and Bowen Zhou. Intuitive fine-tuning: Towards simplifying alignment into a single process. *arXiv preprint arXiv:2405.11870*, 2024.

Maggie Huan, Yuetai Li, Tuney Zheng, Xiaoyu Xu, Seungone Kim, Minxin Du, Radha Poovendran, Graham Neubig, and Xiang Yue. Does math reasoning improve general llm capabilities? understanding transferability of llm reasoning. *arXiv preprint arXiv:2507.00432*, 2025.

Chengsong Huang, Wenhao Yu, Xiaoyang Wang, Hongming Zhang, Zongxia Li, Ruosen Li, Jiaxin Huang, Haitao Mi, and Dong Yu. R-zero: Self-evolving reasoning llm from zero data. *arXiv preprint arXiv:2508.05004*, 2025a.

Shengyi Huang, Rousslan Fernand Julien Dossa, Antonin Raffin, Anssi Kanervisto, and Weixun Wang. The 37 implementation details of proximal policy optimization. *The ICLR Blog Track 2023*, 2022.

Ting Huang, Zeyu Zhang, and Hao Tang. 3d-r1: Enhancing reasoning in 3d vlms for unified scene understanding. *arXiv preprint arXiv:2507.23478*, 2025b.

Wenxuan Huang, Bohan Jia, Zijie Zhai, Shaosheng Cao, Zheyu Ye, Fei Zhao, Zhe Xu, Yao Hu, and Shaohui Lin. Vision-r1: Incentivizing reasoning capability in multimodal large language models. *arXiv preprint arXiv:2503.06749*, 2025c.

Yanxing Huang, Xinling Jin, Sijie Liang, Peng Li, and Yang Liu. Formarl: Enhancing autoformalization with no labeled data. *arXiv preprint arXiv:2508.18914*, 2025d.

Yuzhen Huang, Weihao Zeng, Xingshan Zeng, Qi Zhu, and Junxian He. Pitfalls of rule-and model-based verifiers–a case study on mathematical reasoning. *arXiv preprint arXiv:2505.22203*, 2025e.

Zenan Huang, Yihong Zhuang, Guoshan Lu, Zeyu Qin, Haokai Xu, Tianyu Zhao, Ru Peng, Jiaqi Hu, Zhanming Shen, Xiaomeng Hu, et al. Reinforcement learning with rubric anchors. *arXiv preprint arXiv:2508.12790*, 2025f.

Zeyu Huang, Tianhao Cheng, Zihan Qiu, Zili Wang, Yinghui Xu, Edoardo M Ponti, and Ivan Titov. Blending supervised and reinforcement fine-tuning with prefix sampling. *arXiv preprint arXiv:2507.01679*, 2025g.

Hugging Face. Open r1: A fully open reproduction of deepseek-r1, January 2025. URL https://github.com/huggingface/open-r1.

Dom Huh and Prasant Mohapatra. Multi-agent reinforcement learning: A comprehensive survey. *arXiv preprint arXiv:2312.10256*, 2023.

Jemin Hwangbo, Joonho Lee, Alexey Dosovitskiy, Dario Bellicoso, Vassilios Tsounis, Vladlen Koltun, and Marco Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 4 (26):eaau5872, 2019.

Baichuan Inc. Baichuan-m1: Pushing the medical capability of large language models. *arXiv preprint arXiv::2502.12671*, 2025.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. pi 0.5: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Edward L Ionides. Truncated importance sampling. *Journal of Computational and Graphical Statistics*, 17(2):295–311, 2008.

Ana-Maria Istrate, Fausto Milletari, Fabrizio Castrotorres, Jakub M Tomczak, Michaela Torkar, Donghui Li, and Theofanis Karaletsos. rbio1-training scientific reasoning LLMs with biological world models as soft verifiers. *bioRxiv 2025.08.18.670981*, 2025.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Naman Jain, Jaskirat Singh, Manish Shetty, Liang Zheng, Koushik Sen, and Ion Stoica. R2e-gym: Procedural environments and hybrid verifiers for scaling open-weights swe agents. *arXiv preprint arXiv:2504.07164*, 2025.

Kunal Jha, Wilka Carvalho, Yancheng Liang, Simon Shaolei Du, Max Kleiman-Weiner, and Natasha Jaques. Cross-environment cooperation enables zero-shot multi-agent coordination. In *Forty-second International Conference on Machine Learning*, 2025.

Haozhe Ji, Cheng Lu, Yilin Niu, Pei Ke, Hongning Wang, Jun Zhu, Jie Tang, and Minlie Huang. Towards efficient exact optimization of language model alignment. *arXiv preprint arXiv:2402.00856*, 2024.

Xingguang Ji, Yahui Liu, Qi Wang, Jingyuan Zhang, Yang Yue, Rui Shi, Chenxi Sun, Fuzheng Zhang, Guorui Zhou, and Kun Gai. Leanabell-prover-v2: Verifier-integrated reasoning for formal theorem proving via reinforcement learning. *arXiv preprint arXiv:2507.08649*, 2025a.

Yuheng Ji, Huajie Tan, Jiayu Shi, Xiaoshuai Hao, Yuan Zhang, Hengyuan Zhang, Pengwei Wang, Mengdi Zhao, Yao Mu, Pengju An, et al. Robobrain: A unified brain model for robotic manipulation from abstract to concrete. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1724–1734, 2025b.

Ruipeng Jia, Yunyi Yang, Yongbo Gai, Kai Luo, Shihao Huang, Jianhe Lin, Xiaoxi Jiang, and Guanjun Jiang. Writing-zero: Bridge the gap between non-verifiable tasks and verifiable rewards. *arXiv e-prints*, pages arXiv–2506, 2025.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

Daniel R Jiang, Alex Nikulkov, Yu-Chia Chen, Yang Bai, and Zheqing Zhu. Improving generative ad text on facebook using reinforcement learning. *arXiv preprint arXiv:2507.21983*, 2025a.

Dongzhi Jiang, Ziyu Guo, Renrui Zhang, Zhuofan Zong, Hao Li, Le Zhuo, Shilin Yan, Pheng-Ann Heng, and Hongsheng Li. T2i-r1: Reinforcing image generation with collaborative semantic-level and token-level cot. *arXiv preprint arXiv:2505.00703*, 2025b.

Jingjing Jiang, Chongjie Si, Jun Luo, Hanwang Zhang, and Chao Ma. Co-reinforcement learning for unified multimodal understanding and generation. *arXiv preprint arXiv:2505.17534*, 2025c.

Pengcheng Jiang, Xueqiang Xu, Jiacheng Lin, Jinfeng Xiao, Zifeng Wang, Jimeng Sun, and Jiawei Han. s3: You don't need that much data to train a search agent via rl. *arXiv preprint arXiv:2505.14146*, 2025d.

Yuhua Jiang, Yuwen Xiong, Yufeng Yuan, Chao Xin, Wenyuan Xu, Yu Yue, Qianchuan Zhao, and Lin Yan. Pag: Multi-turn reinforced llm self-correction with policy as generative verifier. *arXiv preprint arXiv:2506.10406*, 2025e.

Yuqian Jiang, Suda Bharadwaj, Bo Wu, Rishi Shah, Ufuk Topcu, and Peter Stone. Temporal-logic-based reward shaping for continuing reinforcement learning tasks. In *Proceedings of the AAAI Conference on artificial Intelligence*, volume 35, pages 7995–8003, 2021.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. Swe-bench: Can language models resolve real-world github issues? *arXiv preprint arXiv:2310.06770*, 2023.

Bowen Jin, Jinsung Yoon, Priyanka Kargupta, Sercan O Arik, and Jiawei Han. An empirical study on reinforcement learning for reasoning-search interleaved llm agents. *arXiv preprint arXiv:2505.15117*, 2025a.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025b.

Can Jin, Yang Zhou, Qixin Zhang, Hongwu Peng, Di Zhang, Marco Pavone, Ligong Han, Zhang-Wei Hong, Tong Che, and Dimitris N Metaxas. Your reward function for rl is your best prm for search: Unifying rl and search-based tts. *arXiv preprint arXiv:2508.14313*, 2025c.

Hangzhan Jin, Sicheng Lv, Sifan Wu, and Mohammad Hamdaqa. Rl is neither a panacea nor a mirage: Understanding supervised vs. reinforcement learning fine-tuning for llms. *arXiv preprint arXiv:2508.16546*, 2025d.

Zhehan Kan, Yanlin Liu, Kun Yin, Xinghua Jiang, Xin Li, Haoyu Cao, Yinsong Liu, Deqiang Jiang, Xing Sun, Qingmin Liao, et al. Taco: Think-answer consistency for optimized long-chain reasoning and efficient data learning via reinforcement learning in lvlms. *arXiv preprint arXiv:2505.20777*, 2025.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. VinePPO: Refining credit assignment in RL training of LLMs. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=Myx2kJFzAn.

Kasia Z. Kedzierska, Lorin Crawford, Ava P. Amini, and Alex X. Lu. Assessing the limits of zero-shot foundation models in single-cell biology, 2023.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. Understanding the effects of rlhf on llm generalisation and diversity. *arXiv preprint arXiv:2310.06452*, 2023.

Andrew Kiruluta, Andreas Lemos, and Priscilla Burity. A self-supervised reinforcement learning approach for fine-tuning large language models using cross-attention signals. *arXiv preprint arXiv:2502.10482*, 2025.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Inception Labs, Samar Khanna, Siddhant Kharbanda, Shufan Li, Harshit Varma, Eric Wang, Sawyer Birnbaum, Ziyang Luo, Yanis Miraoui, Akash Palrecha, et al. Mercury: Ultra-fast language models based on diffusion. *arXiv preprint arXiv:2506.17298*, 2025.

Hanyu Lai, Xiao Liu, Yanxiao Zhao, Han Xu, Hanchen Zhang, Bohao Jing, Yanyu Ren, Shuntian Yao, Yuxiao Dong, and Jie Tang. Computerrl: Scaling end-to-end online reinforcement learning for computer use agents, 2025. URL https://arxiv.org/abs/2508.14040.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896. Atlanta, 2013.

Dong Won Lee, Hae Won Park, Yoon Kim, Cynthia Breazeal, and Louis-Philippe Morency. Improving dialogue agents by decomposing one global explicit annotation with local implicit multimodal feedback. *arXiv preprint arXiv:2403.11330*, 2024a.

Dong Won Lee, Hae Won Park, Cynthia Breazeal, and Louis-Philippe Morency. Aligning dialogue agents with global feedback via large language model reward decomposition. *arXiv preprint arXiv:2505.15922*, 2025.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Ren Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. RLAIF vs. RLHF: Scaling reinforcement learning from human feedback with AI feedback. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 26874–26901. PMLR, 21–27 Jul 2024b. URL https://proceedings.mlr.press/v235/lee24t.html.

Chengpeng Li, Zhengyang Tang, Ziniu Li, Mingfeng Xue, Keqin Bao, Tian Ding, Ruoyu Sun, Benyou Wang, Xiang Wang, Junyang Lin, et al. Cort: Code-integrated reasoning within thinking. *arXiv preprint arXiv:2506.09820*, 2025a.

Chunmao Li, Yang Li, Yinliang Zhao, Peng Peng, and Xupeng Geng. Sler: Self-generated long-term experience replay for continual reinforcement learning. *Applied Intelligence*, 51(1):185–201, 2021.

Derek Li, Jiaming Zhou, Amirreza Kazemi, Qianyi Sun, Abbas Ghaddar, Mohammad Ali Alomrani, Liheng Ma, Yu Luo, Dong Li, Feng Wen, et al. Omni-think: Scaling cross-domain generalization in llms via multi-task rl with hybrid rewards. *arXiv preprint arXiv:2507.14783*, 2025b.

Hao Li, He Cao, Bin Feng, Yanjun Shao, Xiangru Tang, Zhiyuan Yan, Li Yuan, Yonghong Tian, and Yu Li. Beyond chemical qa: Evaluating llm's chemical reasoning with modular chemical operations. *arXiv preprint arXiv:2505.21318*, 2025c.

Hu Li, Xuezhong Qian, and Wei Song. Prioritized experience replay based on dynamics priority. *Scientific Reports*, 14(1):6014, 2024a.

Huao Li, Yu Quan Chong, Simon Stepputtis, Joseph Campbell, Dana Hughes, Michael Lewis, and Katia Sycara. Theory of mind for multi-agent collaboration via large language models. *arXiv preprint arXiv:2310.10701*, 2023a.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, et al. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*, 13:9, 2024b.

Junlong Li, Shichao Sun, Weizhe Yuan, Run-Ze Fan, Hai Zhao, and Pengfei Liu. Generative judge for evaluating alignment. *arXiv preprint arXiv:2310.05470*, 2023b.

Junzhe Li, Yutao Cui, Tao Huang, Yinping Ma, Chun Fan, Miles Yang, and Zhao Zhong. Mixgrpo: Unlocking flow-based grpo efficiency with mixed ode-sde. *arXiv preprint arXiv:2507.21802*, 2025d.

Kuan Li, Zhongwang Zhang, Huifeng Yin, Liwen Zhang, Litu Ou, Jialong Wu, Wenbiao Yin, Baixuan Li, Zhengwei Tao, Xinyu Wang, et al. Websailor: Navigating super-human reasoning for web agent. *arXiv preprint arXiv:2507.02592*, 2025e.

Peiji Li, Jiasheng Ye, Yongkang Chen, Yichuan Ma, Zijie Yu, Kedi Chen, Ganqu Cui, Haozhan Li, Jiacheng Chen, Chengqi Lyu, et al. Internbootcamp technical report: Boosting llm reasoning with verifiable task scaling. *arXiv preprint arXiv:2508.08636*, 2025f.

Pengyi Li, Matvey Skripkin, Alexander Zubrey, Andrey Kuznetsov, and Ivan Oseledets. Confidence is all you need: Few-shot rl fine-tuning of language models. *arXiv preprint arXiv:2506.06395*, 2025g.

Tianjian Li, Yiming Zhang, Ping Yu, Swarnadeep Saha, Daniel Khashabi, Jason Weston, Jack Lanchantin, and Tianlu Wang. Jointly reinforcing diversity and quality in language model generations, 2025h. URL https://arxiv.org/abs/2509.02534.

Weizhen Li, Jianbo Lin, Zhuosong Jiang, Jingyi Cao, Xinpeng Liu, Jiayu Zhang, Zhenqiang Huang, Qianben Chen, Weichen Sun, Qiexiang Wang, et al. Chain-of-agents: End-to-end agent foundation models via multi-agent distillation and agentic rl. *arXiv preprint arXiv:2508.13167*, 2025i.

Wenjie Li, Yujie Zhang, and Haoran Sun. Cx-mind: A pioneering multimodal large language model for interleaved reasoning in chest x-ray via curriculum-guided reinforcement learning, 2025j. URL https://arxiv.org/abs/2508.03733.

Xiaomin Li, Zhou Yu, Zhiwei Zhang, Xupeng Chen, Ziji Zhang, Yingying Zhuang, Narayanan Sadagopan, and Anurag Beniwal. When thinking fails: The pitfalls of reasoning for instruction-following in llms. *arXiv preprint arXiv:2505.11423*, 2025k.

Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yutao Zhu, Yongkang Wu, Ji-Rong Wen, and Zhicheng Dou. Webthinker: Empowering large reasoning models with deep research capability. *arXiv preprint arXiv:2504.21776*, 2025l.

Xingxuan Li, Yao Xiao, Dianwen Ng, Hai Ye, Yue Deng, Xiang Lin, Bin Wang, Zhanfeng Mo, Chong Zhang, Yueyi Zhang, et al. Miromind-m1: An open-source advancement in mathematical reasoning via context-aware multi-stage policy optimization. *arXiv preprint arXiv:2507.14683*, 2025m.

Xinhao Li, Ziang Yan, Desen Meng, Lu Dong, Xiangyu Zeng, Yinan He, Yali Wang, Yu Qiao, Yi Wang, and Limin Wang. Videochat-r1: Enhancing spatio-temporal perception via reinforcement fine-tuning. *arXiv preprint arXiv:2504.06958*, 2025n.

Xuefeng Li, Haoyang Zou, and Pengfei Liu. Limr: Less is more for rl scaling. *arXiv preprint arXiv:2502.11886*, 2025o.

Xuefeng Li, Haoyang Zou, and Pengfei Liu. Torl: Scaling tool-integrated rl. *arXiv preprint arXiv:2503.23383*, 2025p.

Yizhi Li, Qingshui Gu, Zhoufutu Wen, Ziniu Li, Tianshun Xing, Shuyue Guo, Tianyu Zheng, Xin Zhou, Xingwei Qu, Wangchunshu Zhou, Zheng Zhang, Wei Shen, Qian Liu, Chenghua Lin, Jian Yang, Ge Zhang, and Wenhao Huang. Treepo: Bridging the gap of policy optimization and efficacy and inference efficiency with heuristic tree-based modeling. *arXiv preprint arXiv:2508.17445*, 2025q.

Yue Li, Meng Tian, Dechang Zhu, Jiangtong Zhu, Zhenyu Lin, Zhiwei Xiong, and Xinhai Zhao. Drive-r1: Bridging reasoning and planning in vlms for autonomous driving with reinforcement learning. *arXiv preprint arXiv:2506.18234*, 2025r.

Yuetai Li, Zhangchen Xu, Fengqing Jiang, Bhaskar Ramasubramanian, Luyao Niu, Bill Yuchen Lin, Xiang Yue, and Radha Poovendran. Temporal sampling for forgotten reasoning in llms. *arXiv preprint arXiv:2505.20196*, 2025s.

Zaijing Li, Yuquan Xie, Rui Shao, Gongwei Chen, Weili Guan, Dongmei Jiang, and Liqiang Nie. Optimus-3: Towards generalist multimodal minecraft agents with scalable task experts. *arXiv preprint arXiv:2506.10357*, 2025t.

Zhong-Zhi Li, Duzhen Zhang, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian Xu, Junhao Zheng, Pei-Jie Wang, Xiuyi Chen, et al. From system 1 to system 2: A survey of reasoning large language models. *arXiv preprint arXiv:2502.17419*, 2025u.

Ziniu Li, Tian Xu, Yushun Zhang, Zhihang Lin, Yang Yu, Ruoyu Sun, and Zhi-Quan Luo. Remax: A simple, effective, and efficient reinforcement learning method for aligning large language models. *arXiv preprint arXiv:2310.10505*, 2023c.

Jing Liang, Hongyao Tang, Yi Ma, Jinyi Liu, Yan Zheng, Shuyue Hu, Lei Bai, and Jianye Hao. Squeeze the soaked sponge: Efficient off-policy reinforcement finetuning for large language model. *arXiv preprint arXiv:2507.06892*, 2025a.

Xiao Liang, Zhong-Zhi Li, Yeyun Gong, Yang Wang, Hengyuan Zhang, Yelong Shen, Ying Nian Wu, and Weizhu Chen. Sws: Self-aware weakness-driven problem synthesis in reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.08989*, 2025b.

Xiao Liang, Zhongzhi Li, Yeyun Gong, Yelong Shen, Ying Nian Wu, Zhijiang Guo, and Weizhu Chen. Beyond pass@ 1: Self-play with variational problem synthesis sustains rlvr. *arXiv preprint arXiv:2508.14029*, 2025c.

Zhanhao Liang, Yuhui Yuan, Shuyang Gu, Bohan Chen, Tiankai Hang, Mingxi Cheng, Ji Li, and Liang Zheng. Aesthetic post-training diffusion models from generic preferences with step-by-step preference optimization. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 13199–13208, 2025d.

Jianxing Liao, Tian Zhang, Xiao Feng, Yusong Zhang, Rui Yang, Haorui Wang, Bosi Wen, Ziying Wang, and Runzhi Shi. Rlmr: Reinforcement learning with mixed rewards for creative writing. *arXiv preprint arXiv:2508.18642*, 2025a.

Mengqi Liao, Xiangyu Xi, Ruinian Chen, Jia Leng, Yangen Hu, Ke Zeng, Shuai Liu, and Huaiyu Wan. Enhancing efficiency and exploration in reinforcement learning for llms. *arXiv preprint arXiv:2505.18573*, 2025b.

Zhenyi Liao, Qingsong Xie, Yanhao Zhang, Zijian Kong, Haonan Lu, Zhenyu Yang, and Zhijie Deng. Improved visual-spatial reasoning via r1-zero-like training. *arXiv preprint arXiv:2504.00883*, 2025c.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Darryl Lin, Sachin Talathi, and Sreekanth Annapureddy. Fixed point quantization of deep convolutional networks. In *International conference on machine learning*, pages 2849–2858. PMLR, 2016.

Hongyu Lin, Yuchen Li, Haoran Luo, Kaichun Yao, Libo Zhang, Mingjie Xing, and Yanjun Wu. Os-r1: Agentic operating system kernel tuning with reinforcement learning. *arXiv preprint arXiv:2508.12551*, 2025a.

Jiacheng Lin and Zhenbang Wu. Training llms for ehr-based reasoning tasks via reinforcement learning, 2025. URL https://arxiv.org/abs/2505.24105.

Wang Lin, Liyu Jia, Wentao Hu, Kaihang Pan, Zhongqi Yue, Wei Zhao, Jingyuan Chen, Fei Wu, and Hanwang Zhang. Reasoning physical video generation with diffusion timestep tokens via reinforcement learning. *arXiv preprint arXiv:2504.15932*, 2025b.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Bo Liu, Leon Guertler, Simon Yu, Zichen Liu, Penghui Qi, Daniel Balcells, Mickel Liu, Cheston Tan, Weiyan Shi, Min Lin, et al. Spiral: Self-play on zero-sum games incentivizes reasoning via multi-agent multi-turn reinforcement learning. *arXiv preprint arXiv:2506.24119*, 2025a.

Fangfu Liu, Hanyang Wang, Yimo Cai, Kaiyan Zhang, Xiaohang Zhan, and Yueqi Duan. Video-t1: Test-time scaling for video generation. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2025b.

Jia Liu, ChangYi He, YingQiao Lin, MingMin Yang, FeiYang Shen, ShaoGuo Liu, and TingTing Gao. Ettrl: Balancing exploration and exploitation in llm test-time reinforcement learning via entropy mechanism. *arXiv preprint arXiv:2508.11356*, 2025c.

Jiawei Liu and Lingming Zhang. Code-r1: Reproducing r1 for code with reliable rewards. https://github.com/ganler/code-r1, 2025.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. *arXiv preprint arXiv:2505.05470*, 2025d.

Jie Liu, Gongye Liu, Jiajun Liang, Ziyang Yuan, Xiaokun Liu, Mingwu Zheng, Xiele Wu, Qiulin Wang, Wenyu Qin, Menghan Xia, et al. Improving video generation with human feedback. *arXiv preprint arXiv:2501.13918*, 2025e.

Jijia Liu, Feng Gao, Bingwen Wei, Xinlei Chen, Qingmin Liao, Yi Wu, Chao Yu, and Yu Wang. What can rl bring to vla generalization? an empirical study. *arXiv preprint arXiv:2505.19789*, 2025f.

Junteng Liu, Yuanxiang Fan, Zhuo Jiang, Han Ding, Yongyi Hu, Chi Zhang, Yiqi Shi, Shitong Weng, Aili Chen, Shiqi Chen, et al. Synlogic: Synthesizing verifiable reasoning data at scale for learning logical reasoning and beyond. *arXiv preprint arXiv:2505.19641*, 2025g.

Lijun Liu and Ruiyang Li. Efficient medical vie via reinforcement learning, 2025. URL https://arxiv.org/abs/2506.13363.

Liyuan Liu, Feng Yao, Dinghuai Zhang, Chengyu Dong, Jingbo Shang, and Jianfeng Gao. Flashrl: 8bit rollouts, full power rl, August 2025h. URL https://fengyao.notion.site/flash-rl.

Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. Prorl: Prolonged reinforcement learning expands reasoning boundaries in large language models. *arXiv preprint arXiv:2505.24864*, 2025i.

Mingyang Liu, Gabriele Farina, and Asuman Ozdaglar. Uft: Unifying supervised and reinforcement fine-tuning. *arXiv preprint arXiv:2505.16984*, 2025j.

Runtao Liu, Haoyu Wu, Ziqiang Zheng, Chen Wei, Yingqing He, Renjie Pi, and Qifeng Chen. Videodpo: Omni-preference alignment for video diffusion generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 8009–8019, 2025k.

Runze Liu, Fengshuo Bai, Yali Du, and Yaodong Yang. Meta-reward-net: Implicitly differentiable reward learning for preference-based reinforcement learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 22270–22284. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/8be9c134bb193d8bd3827d4df8488228-Paper-Conference.pdf.

Runze Liu, Junqi Gao, Jian Zhao, Kaiyan Zhang, Xiu Li, Biqing Qi, Wanli Ouyang, and Bowen Zhou. Can 1b llm surpass 405b llm? rethinking compute-optimal test-time scaling. *arXiv preprint arXiv:2502.06703*, 2025l.

Shudong Liu, Hongwei Liu, Junnan Liu, Linchen Xiao, Songyang Gao, Chengqi Lyu, Yuzhe Gu, Wenwei Zhang, Derek F Wong, Songyang Zhang, et al. Compassverifier: A unified and robust verifier for llms evaluation and outcome reward. *arXiv preprint arXiv:2508.03686*, 2025m.

Shuo Liu, Zeyu Liang, Xueguang Lyu, and Christopher Amato. Llm collaboration with multi-agent reinforcement learning. *arXiv preprint arXiv:2508.04652*, 2025n.

Tianqi Liu, Yao Zhao, Rishabh Joshi, Misha Khalman, Mohammad Saleh, Peter J Liu, and Jialu Liu. Statistical rejection sampling improves preference optimization. *arXiv preprint arXiv:2309.06657*, 2023a.

Wei Liu, Ruochen Zhou, Yiyun Deng, Yuzhen Huang, Junteng Liu, Yuntian Deng, Yizhe Zhang, and Junxian He. Learn to reason efficiently with adaptive length-based reward shaping. *arXiv preprint arXiv:2505.15612*, 2025o.

Xingchao Liu, Chengyue Gong, et al. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023b.

Yifei Liu, Li Lyna Zhang, Yi Zhu, Bingcheng Dong, Xudong Zhou, Ning Shang, Fan Yang, and Mao Yang. rstar-coder: Scaling competitive code reasoning with a large-scale verified dataset. *arXiv preprint arXiv:2505.21297*, 2025p.

Yizhou Liu and Jingwei Wei. Breaking reward collapse: Adaptive reinforcement for open-ended medical reasoning with enhanced semantic discrimination, 2025. URL https://arxiv.org/ab s/2508.12957.

Yuhang Liu, Pengxiang Li, Congkai Xie, Xavier Hu, Xiaotian Han, Shengyu Zhang, Hongxia Yang, and Fei Wu. Infigui-r1: Advancing multimodal gui agents from reactive actors to deliberative reasoners. *arXiv preprint arXiv:2504.14239*, 2025q.

Zexi Liu, Jingyi Chai, Xinyu Zhu, Shuo Tang, Rui Ye, Bo Zhang, Lei Bai, and Siheng Chen. Ml-agent: Reinforcing llm agents for autonomous machine learning engineering. *arXiv preprint arXiv:2505.23723*, 2025r.

Zichen Liu, Changyu Chen, Wenjun Li, Tianyu Pang, Chao Du, and Min Lin. There may not be aha moment in r1-zero-like training—a pilot study, 2025s.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025t.

Zihan Liu, Zhuolin Yang, Yang Chen, Chankyu Lee, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. Acereason-nemotron 1.1: Advancing math and code reasoning through sft and rl synergy. *arXiv preprint arXiv:2506.13284*, 2025u.

Zihe Liu, Jiashun Liu, Yancheng He, Weixun Wang, Jiaheng Liu, Ling Pan, Xinyu Hu, Shaopan Xiong, Ju Huang, Jian Hu, et al. Part i: Tricks or traps? a deep dive into rl for llm reasoning. *arXiv preprint arXiv:2508.08221*, 2025v.

Zijun Liu, Peiyi Wang, Runxin Xu, Shirong Ma, Chong Ruan, Peng Li, Yang Liu, and Yu Wu. Inference-time scaling for generalist reward modeling. *arXiv preprint arXiv:2504.02495*, 2025w.

Ziyu Liu, Zeyi Sun, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Haodong Duan, Dahua Lin, and Jiaqi Wang. Visual-rft: Visual reinforcement fine-tuning. *arXiv preprint arXiv:2503.01785*, 2025x.

Zongkai Liu, Fanqing Meng, Lingxiao Du, Zhixiang Zhou, Chao Yu, Wenqi Shao, and Qiaosheng Zhang. Cpgd: Toward stable rule-based reinforcement learning for language models. *arXiv preprint arXiv:2505.12504*, 2025y.

Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30, 2017.

Dakuan Lu, Xiaoyu Tan, Rui Xu, Tianchu Yao, Chao Qu, Wei Chu, Yinghui Xu, and Yuan Qi. Scp-116k: A high-quality problem-solution dataset and a generalized pipeline for automated extraction in the higher education science domain. *arXiv preprint arXiv:2501.15587*, 2025a.

Fanbin Lu, Zhisheng Zhong, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Arpo: End-to-end policy optimization for gui agents with experience replay. *arXiv preprint arXiv:2505.16282*, 2025b.

Guanxing Lu, Wenkai Guo, Chubin Zhang, Yuheng Zhou, Haonan Jiang, Zifeng Gao, Yansong Tang, and Ziwei Wang. Vla-rl: Towards masterful and general robotic manipulation with scalable reinforcement learning. *arXiv preprint arXiv:2505.18719*, 2025c.

Jianqiao Lu, Zhiyang Dou, Hongru Wang, Zeyu Cao, Jianbo Dai, Yunlong Feng, and Zhijiang Guo. Autopsv: Automated process-supervised verifier. *Advances in Neural Information Processing Systems*, 37:79935–79962, 2024.

Quanfeng Lu, Zhantao Ma, Shuai Zhong, Jin Wang, Dahai Yu, Michael K Ng, and Ping Luo. Swirl: A staged workflow for interleaved reinforcement learning in mobile gui control. *arXiv preprint arXiv:2508.20018*, 2025d.

Songshuo Lu, Hua Wang, Zhi Chen, and Yaohua Tang. Urpo: A unified reward & policy optimization framework for large language models. *arXiv preprint arXiv:2507.17515*, 2025e.

Zhengxi Lu, Yuxiang Chai, Yaxuan Guo, Xi Yin, Liang Liu, Hao Wang, Han Xiao, Shuai Ren, Guanjing Xiong, and Hongsheng Li. Ui-r1: Enhancing efficient action prediction of gui agents by reinforcement learning. *arXiv preprint arXiv:2503.21620*, 2025f.

Fan-Ming Luo, Tian Xu, Hang Lai, Xiong-Hui Chen, Weinan Zhang, and Yang Yu. A survey on model-based reinforcement learning. *Science China Information Sciences*, 67(2):121101, 2024.

Haotian Luo, Li Shen, Haiying He, Yibo Wang, Shiwei Liu, Wei Li, Naiqiang Tan, Xiaochun Cao, and Dacheng Tao. O1-pruner: Length-harmonizing fine-tuning for o1-like reasoning pruning. *arXiv preprint arXiv:2501.12570*, 2025a.

Michael Luo, Sijun Tan, Roy Huang, Ameen Patel, Alpay Ariyak, Qingyang Wu, Xiaoxiang Shi, Rachel Xin, Colin Cai, Maurice Weber, Ce Zhang, Li Erran Li, Raluca Ada Popa, and Ion Stoica. Deepcoder: A fully open-source 14b coder at o3-mini level. https://pretty-radio-b75.notion.site/DeepCoder-A-Fully-Open-Source-14B-Coder-at-O3-mini-Level-1cf81902c14680b3bee5eb349a512a51, 2025b. Notion Blog.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Li Erran Li, et al. Deepscaler: Surpassing o1-preview with a 1.5 b model by scaling rl. *Notion Blog*, 2025c.

Run Luo, Lu Wang, Wanwei He, and Xiaobo Xia. Gui-r1: A generalist r1-style vision-language action model for gui agents. *arXiv preprint arXiv:2504.10458*, 2025d.

Xufang Luo, Yuge Zhang, Zhiyuan He, Zilong Wang, Siyun Zhao, Dongsheng Li, Luna K Qiu, and Yuqing Yang. Agent lightning: Train any ai agents with reinforcement learning. *arXiv preprint arXiv:2508.03680*, 2025e.

Xingtai Lv, Yuxin Zuo, Youbang Sun, Hongyi Liu, Yuntian Wei, Zhekai Chen, Lixuan He, Xuekai Zhu, Kaiyan Zhang, Bingning Wang, et al. Towards a unified view of large language model post-training. *arXiv preprint arXiv:2509.04419*, 2025.

Chengqi Lyu, Songyang Gao, Yuzhe Gu, Wenwei Zhang, Jianfei Gao, Kuikun Liu, Ziyi Wang, Shuaibin Li, Qian Zhao, Haian Huang, et al. Exploring the limit of outcome reward for learning mathematical reasoning. *arXiv preprint arXiv:2502.06781*, 2025.

Lu Ma, Hao Liang, Meiyi Qiang, Lexiang Tang, Xiaochen Ma, Zhen Hao Wong, Junbo Niu, Chengyu Shen, Runming He, Bin Cui, et al. Learning what reinforcement learning can't: Interleaved online fine-tuning for hardest questions. *arXiv preprint arXiv:2506.07527*, 2025a.

Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, et al. Inference-time scaling for diffusion models beyond scaling denoising steps. *arXiv preprint arXiv:2501.09732*, 2025b.

Xueguang Ma, Qian Liu, Dongfu Jiang, Ge Zhang, Zejun Ma, and Wenhu Chen. General-reasoner: Advancing llm reasoning across all domains. *arXiv preprint arXiv:2505.14652*, 2025c.

Dakota Mahan, Duy Van Phung, Rafael Rafailov, Chase Blagden, Nathan Lile, Louis Castricato, Jan-Philipp Fränken, Chelsea Finn, and Alon Albalak. Generative reward models. *arXiv preprint arXiv:2410.12832*, 2024.

Justus Mattern, Sami Jaghouar, Manveer Basra, Jannik Straube, Matthew Di Ferrante, Felix Gabriel, Jack Min Ong, Vincent Weisser, and Johannes Hagemann. Synthetic-1: Two million collaboratively generated reasoning traces from deepseek-r1, 2025. URL https://www.primeintellect.ai/blog/synthetic-1-release.

Jianbiao Mei, Tao Hu, Daocheng Fu, Licheng Wen, Xuemeng Yang, Rong Wu, Pinlong Cai, Xinyu Cai, Xing Gao, Yu Yang, et al. O2-searcher: A searching-based agent model for open-domain open-ended question answering. *arXiv preprint arXiv:2505.16582*, 2025.

Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. Gaia: a benchmark for general ai assistants. In *The Twelfth International Conference on Learning Representations*, 2023.

Thomas M Moerland, Joost Broekens, Aske Plaat, Catholijn M Jonker, et al. Model-based reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 16(1):1–118, 2023.

Ivan Moshkov, Darragh Hanley, Ivan Sorokin, Shubham Toshniwal, Christof Henkel, Benedikt Schifferer, Wei Du, and Igor Gitman. Aimo-2 winning solution: Building state-of-the-art mathematical reasoning models with openmathreasoning dataset. *arXiv preprint arXiv:2504.16891*, 2025.

Sagnik Mukherjee, Lifan Yuan, Dilek Hakkani-Tur, and Hao Peng. Reinforcement learning finetunes small subnetworks in large language models. *arXiv preprint arXiv:2505.11711*, 2025.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

Jaehyun Nam, Jinsung Yoon, Jiefeng Chen, Jinwoo Shin, Sercan Ö Arık, and Tomas Pfister. Mlestar: Machine learning engineering agent via search and targeted refinement. *arXiv preprint arXiv:2506.15692*, 2025.

Siddharth M. Narayanan, James D. Braza, Ryan-Rhys Griffiths, Albert Bou, Geemi Wellawatte, Mayk Caldas Ramos, Ludovico Mitchener, Samuel G. Rodriques, and Andrew D. White. Training a scientific reasoning model for chemistry. *arXiv preprint arXiv: 2506.17238*, 2025.

Deepak Nathani, Lovish Madaan, Nicholas Roberts, Nikolay Bashlykov, Ajay Menon, Vincent Moens, Amar Budhiraja, Despoina Magka, Vladislav Vorotilov, Gaurav Chaurasia, et al. Mlgym: A new framework and benchmark for advancing ai research agents. *arXiv preprint arXiv:2502.14499*, 2025.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models, 2025. URL https://arxiv.org/abs/2502.09992.

Datta Nimmaturi, Vaishnavi Bhargava, Rajat Ghosh, Johnu George, and Debojyoti Dutta. Predictive scaling laws for efficient grpo training of large reasoning models. *arXiv preprint arXiv:2507.18014*, 2025.

Emmanuel Noutahi, Jason Hartford, Prudencio Tossou, Shawn Whitfield, Alisandra K. Denton, Cas Wognum, Kristina Ulicna, Michael Craig, Jonathan Hsu, Michael Cuccarese, et al. Virtual cells: Predict, explain, discover, 2025.

Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco JR Ruiz, Abbas Mehrabian, et al. Alphaevolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint arXiv:2506.13131*, 2025.

Humza Nusrat. Autonomous radiotherapy treatment planning using dola: A privacy-preserving, llm-based optimization agent, 2025. URL https://arxiv.org/abs/2503.17553.

NVIDIA-NeMo. Nemo rl: A scalable and efficient post-training library. https://github.com/NVIDIA-NeMo/RL, 2025. GitHub repository.

Hayeon Oh. Laviplan: Language-guided visual path planning with rlvr. *arXiv preprint arXiv:2507.12911*, 2025.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, et al. 2 olmo 2 furious. *arXiv preprint arXiv:2501.00656*, 2024.

OpenAI. Introducing gpt-4o image generation. https://openai.com/index/introducing-4o-image-generation/, 2024. Accessed: 2025-08-25.

OpenAI. Gpt-5 system card. *Blog*, 2025a.

OpenAI. Openai o3 and o4-mini system card. *Blog*, 2025b.

Kun Ouyang, Yuanxin Liu, Haoning Wu, Yi Liu, Hao Zhou, Jie Zhou, Fandong Meng, and Xu Sun. Spacer: Reinforcing mllms in video spatial reasoning. *arXiv preprint arXiv:2504.01805*, 2025.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024.

Chaofan Pan, Xin Yang, Yanhua Li, Wei Wei, Tianrui Li, Bo An, and Jiye Liang. A survey of continual reinforcement learning. *arXiv preprint arXiv:2506.21872*, 2025a.

Jiadong Pan, Zhiyuan Ma, Kaiyan Zhang, Ning Ding, and Bowen Zhou. Self-reflective reinforcement learning for diffusion-based image reasoning generation. *arXiv preprint arXiv:2505.22407*, 2025b.

Jiayi Pan, Xingyao Wang, Graham Neubig, Navdeep Jaitly, Heng Ji, Alane Suhr, and Yizhe Zhang. Training software engineering agents and verifiers with swe-gym. *arXiv preprint arXiv:2412.21139*, 2024.

Jiayi Pan, Junjie Zhang, Xingyao Wang, Lifan Yuan, Hao Peng, and Alane Suhr. Tinyzero. https://github.com/Jiayi-Pan/TinyZero, 2025c. Accessed: 2025-01-24.

Jiazhen Pan, Che Liu, and Junde Wu. Medvlm-r1: Incentivizing medical reasoning capability of vision-language models (vlms) via reinforcement learning, 2025d. URL https://arxiv.org/abs/2502.19634.

Kaihang Pan, Wendong Bu, Yuruo Wu, Yang Wu, Kai Shen, Yunfei Li, Hang Zhao, Juncheng Li, Siliang Tang, and Yueting Zhuang. Focusdiff: Advancing fine-grained text-image alignment for autoregressive visual generation through rl. *arXiv preprint arXiv:2506.05501*, 2025e.

Zhenyu Pan and Han Liu. Metaspatial: Reinforcing 3d spatial reasoning in vlms for the metaverse. *arXiv preprint arXiv:2503.18470*, 2025.

Shubham Parashar, Shurui Gui, Xiner Li, Hongyi Ling, Sushil Vemuri, Blake Olson, Eric Li, Yu Zhang, James Caverlee, Dileep Kalathil, et al. Curriculum reinforcement learning from easy to hard tasks improves llm reasoning. *arXiv preprint arXiv:2506.06632*, 2025.

Chanwoo Park, Seungju Han, Xingzhi Guo, Asuman Ozdaglar, Kaiqing Zhang, and Joo-Kyung Kim. Maporl: Multi-agent post-co-training for collaborative large language models with reinforcement learning. *arXiv preprint arXiv:2502.18439*, 2025.

Jupinder Parmar, Sanjev Satheesh, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Reuse, don't retrain: A recipe for continued pretraining of language models. *arXiv preprint arXiv:2407.07263*, 2024.

Guilherme Penedo, Anton Lozhkov, Hynek Kydlíček, Loubna Ben Allal, Edward Beeching, Agustín Piqueres Lajarín, Quentin Gallouédec, Nathan Habib, Lewis Tunstall, and Leandro von Werra. Codeforces cots. https://huggingface.co/datasets/open-r1/codeforces-cots, 2025.

Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel Van de Panne. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions On Graphics (TOG)*, 37(4):1–14, 2018.

Julien Perolat, Bart De Vylder, Daniel Hennes, Eugene Tarassov, Florian Strub, Vincent de Boer, Paul Muller, Jerome T Connor, Neil Burch, Thomas Anthony, et al. Mastering the game of stratego with model-free multiagent reinforcement learning. *Science*, 378(6623):990–996, 2022.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, and et al. Humanity's last exam, 2025. URL https://arxiv.org/abs/2501.14249.

Gabriel Poesia, David Broman, Nick Haber, and Noah Goodman. Learning formal mathematics from intrinsic motivation. *Advances in Neural Information Processing Systems*, 37:43032–43057, 2024.

Mohammadreza Pourreza, Shayan Talaei, Ruoxi Sun, Xingchen Wan, Hailong Li, Azalia Mirhoseini, Amin Saberi, Sercan Arik, et al. Reasoning-sql: Reinforcement learning with sql tailored partial rewards for reasoning-enhanced text-to-sql. *arXiv preprint arXiv:2503.23157*, 2025.

Mihir Prabhudesai, Lili Chen, Alex Ippoliti, Katerina Fragkiadaki, Hao Liu, and Deepak Pathak. Maximizing confidence alone improves reasoning. *arXiv preprint arXiv:2505.22660*, 2025.

Ori Press, Ravid Shwartz-Ziv, Yann LeCun, and Matthias Bethge. The entropy enigma: Success and failure of entropy minimization. *arXiv preprint arXiv:2405.05012*, 2024.

PrimeIntellect. Synthetic-2 release: Four million collaboratively generated reasoning traces. `https://www.primeintellect.ai/blog/synthetic-2-release#synthetic-2-dataset`, 2025. Technical Report.

Zehan Qi, Xiao Liu, Iat Long Iong, Hanyu Lai, Xueqiao Sun, Wenyi Zhao, Yu Yang, Xinyue Yang, Jiadai Sun, Shuntian Yao, et al. Webrl: Training llm web agents via self-evolving online curriculum reinforcement learning. *arXiv preprint arXiv:2411.02337*, 2024.

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. Toolrl: Reward is all tool learning needs. *arXiv preprint arXiv:2504.13958*, 2025.

Rushi Qiang, Yuchen Zhuang, Yinghao Li, Rongzhi Zhang, Changhao Li, Ian Shu-Hei Wong, Sherry Yang, Percy Liang, Chao Zhang, Bo Dai, et al. Mle-dojo: Interactive environments for empowering llm agents in machine learning engineering. *arXiv preprint arXiv:2505.07782*, 2025.

Chongli Qin and Jost Tobias Springenberg. Supervised fine tuning on curated data is reinforcement learning (and can be improved). *arXiv preprint arXiv:2507.12856*, 2025.

Yujia Qin, Yining Ye, Junjie Fang, Haoming Wang, Shihao Liang, Shizuo Tian, Junda Zhang, Jiahao Li, Yunxin Li, Shijue Huang, et al. Ui-tars: Pioneering automated gui interaction with native agents. *arXiv preprint arXiv:2501.12326*, 2025.

Zhongxi Qiu, Zhang Zhang, Yan Hu, Heng Li, and Jiang Liu. Open-medical-r1: How to choose data for rlvr training at medicine domain, 2025. URL `https://arxiv.org/abs/2504.13950`.

Zipeng Qiu. Opentable-r1: A reinforcement learning augmented tool agent for open-domain table question answering. *arXiv preprint arXiv:2507.03018*, 2025.

Xiaoye Qu, Yafu Li, Zhaochen Su, Weigao Sun, Jianhao Yan, Dongrui Liu, Ganqu Cui, Daizong Liu, Shuxian Liang, Junxian He, et al. A survey of efficient reasoning for large reasoning models: Language, multimodality, and beyond. *arXiv preprint arXiv:2503.21614*, 2025a.

Yuxiao Qu, Matthew Y. R. Yang, Amrith Setlur, Lewis Tunstall, Edward Emanuel Beeching, Ruslan Salakhutdinov, and Aviral Kumar. Optimizing test-time compute via meta reinforcement finetuning. In *Forty-second International Conference on Machine Learning*, 2025b. URL `https://openreview.net/forum?id=TqODUDsU4u`.

Qwen Team. Qvq: To see the world with wisdom, 2025. URL `https://qwenlm.github.io/blog/qvq-72b-preview`.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Rafael Rafailov, Joey Hejna, Ryan Park, and Chelsea Finn. From r to q star: Your language model is secretly a q-function. In *First Conference on Language Modeling*, 2024. URL `https://openreview.net/forum?id=kEVcNxtqXk`.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Monotonic value function factorisation for deep multi-agent reinforcement learning. *Journal of Machine Learning Research*, 21(178):1–51, 2020.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining,* pages 3505–3506, 2020.

Abhinav Rastogi, Albert Q. Jiang, Andy Lo, Gabrielle Berrada, Guillaume Lample, Jason Rute, Joep Barmentlo, Karmesh Yadav, Kartik Khandelwal, Khyathi Raghavi Chandu, et al. Magistral. *arxiv preprint arXiv: 2506.10910*, 2025.

ZZ Ren, Zhihong Shao, Junxiao Song, Huajian Xin, Haocheng Wang, Wanjia Zhao, Liyue Zhang, Zhe Fu, Qihao Zhu, Dejian Yang, et al. Deepseek-prover-v2: Advancing formal mathematical reasoning via reinforcement learning for subgoal decomposition. *arXiv preprint arXiv:2504.21801*, 2025.

Syed Asad Rizvi, Daniel Levine, Aakash Patel, Shiyang Zhang, Eric Wang, Sizhuang He, David Zhang, Cerise Tang, Zhuoyang Lyu, Rayyan Darji, Chang Li, Emily Sun, David Jeong, Lawrence Zhao, Jennifer Kwan, David Braun, Brian Hafler, Jeffrey Ishizuka, Rahul M Dhodapkar, Hattie Chung, Shekoofeh Azizi, Bryan Perozzi, and David van Dijk. Scaling large language models for next-generation single-cell analysis. *bioRxiv: 2025.04.14.648850*, 2025.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in neural information processing systems*, 32, 2019.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

Nicolas Le Roux, Marc G Bellemare, Jonathan Lebensold, Arnaud Bergeron, Joshua Greaves, Alex Fréchette, Carolyne Pelletier, Eric Thibodeau-Laufer, Sándor Toth, and Sam Work. Tapered off-policy reinforce: Stable and efficient reinforcement learning for llms. *arXiv preprint arXiv:2503.14286*, 2025.

Lloyd Russell, Anthony Hu, Lorenzo Bertoni, George Fedoseev, Jamie Shotton, Elahe Arani, and Gianluca Corrado. Gaia-2: A controllable multi-view generative world model for autonomous driving. *arXiv preprint arXiv:2503.20523*, 2025.

Ranjan Sapkota, Yang Cao, Konstantinos I Roumeliotis, and Manoj Karkee. Vision-language-action models: Concepts, progress, applications and challenges. *arXiv preprint arXiv:2505.04769*, 2025.

Ali Satvaty, Suzan Verberne, and Fatih Turkmen. Undesirable memorization in large language models: A survey. *arXiv preprint arXiv:2410.02650*, 2024.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551, 2023.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015a.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015b.

John Schulman, Xi Chen, and Pieter Abbeel. Equivalence between policy gradients and soft q-learning. *arXiv preprint arXiv:1704.06440*, 2017a.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017b.

ByteDance Seed, Jiaze Chen, Tiantian Fan, Xin Liu, Lingjun Liu, Zhiqi Lin, Mingxuan Wang, Chengyi Wang, Xiangpeng Wei, Wenyuan Xu, et al. Seed1. 5-thinking: Advancing superb reasoning models with reinforcement learning. *arXiv preprint arXiv:2504.13914*, 2025a.

ByteDance Seed, Jiaze Chen, Tiantian Fan, Xin Liu, Lingjun Liu, Zhiqi Lin, Mingxuan Wang, Chengyi Wang, Xiangpeng Wei, Wenyuan Xu, et al. Seed1.5-thinking: Advancing superb reasoning models with reinforcement learning, 2025b.

Andrew Sellergren, Sahar Kazemzadeh, Tiam Jaroensri, Atilla Kiraly, Madeleine Traverse, Timo Kohlberger, Shawn Xu, Fayaz Jamil, Cían Hughes, Charles Lau, et al. Medgemma technical report. *arXiv preprint arXiv:2507.05201*, 2025.

Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. Rewarding progress: Scaling automated process verifiers for llm reasoning. *arXiv preprint arXiv:2410.08146*, 2024.

Amrith Setlur, Matthew YR Yang, Charlie Snell, Jeremy Greer, Ian Wu, Virginia Smith, Max Simchowitz, and Aviral Kumar. e3: Learning to explore enables extrapolation of test-time compute for llms. *arXiv preprint arXiv:2506.09026*, 2025.

Zeyang Sha, Shiwen Cui, and Weiqiang Wang. Sem: Reinforcement learning for search-efficient large language models. *arXiv preprint arXiv:2505.07903*, 2025.

Sheikh Shafayat, Fahim Tajwar, Ruslan Salakhutdinov, Jeff Schneider, and Andrea Zanette. Can large reasoning models self-train? *arXiv preprint arXiv:2505.21444*, 2025.

Shijie Shang, Ruosi Wan, Yue Peng, Yutong Wu, Xiong-hui Chen, Jie Yan, and Xiangyu Zhang. Stepfun-prover preview: Let's think and verify step by step. *arXiv preprint arXiv:2507.20199*, 2025.

Rulin Shao, Shuyue Stella Li, Rui Xin, Scott Geng, Yiping Wang, Sewoong Oh, Simon Shaolei Du, Nathan Lambert, Sewon Min, Ranjay Krishna, et al. Spurious rewards: Rethinking training signals in rlvr. *arXiv preprint arXiv:2506.10947*, 2025.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, et al. Vlm-r1: A stable and generalizable r1-style large vision-language model. *arXiv preprint arXiv:2504.07615*, 2025a.

Wei Shen, Jiangbo Pei, Yi Peng, Xuchen Song, Yang Liu, Jian Peng, Haofeng Sun, Yunzhuo Hao, Peiyu Wang, Jianhao Zhang, et al. Skywork-r1v3 technical report. *arXiv preprint arXiv:2507.06167*, 2025b.

Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. Rl's razor: Why online reinforcement learning forgets less. *arXiv preprint arXiv:2509.04259*, 2025.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pages 1279–1297, 2025.

Jiajun Shi, Jian Yang, Jiaheng Liu, Xingyuan Bu, Jiangjie Chen, Junting Zhou, Kaijing Ma, Zhoufutu Wen, Bingli Wang, Yancheng He, et al. Korgym: A dynamic game platform for llm reasoning evaluation. *arXiv preprint arXiv:2505.14552*, 2025a.

Taiwei Shi, Yiyang Wu, Linxin Song, Tianyi Zhou, and Jieyu Zhao. Efficient reinforcement finetuning via adaptive curriculum learning. *arXiv preprint arXiv:2504.05520*, 2025b.

Yucheng Shi, Wenhao Yu, Zaitang Li, Yonglin Wang, Hongming Zhang, Ninghao Liu, Haitao Mi, and Dong Yu. Mobilegui-rl: Advancing mobile gui agent through reinforcement learning in online environment. *arXiv preprint arXiv:2507.05720*, 2025c.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Cote, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations*, 2020.

Vaishnavi Shrivastava, Ahmed Awadallah, Vidhisha Balachandran, Shivam Garg, Harkirat Behl, and Dimitris Papailiopoulos. Sample more to think less: Group filtered policy optimization for concise reasoning. *arXiv preprint arXiv:2508.09726*, 2025.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. The curse of recursion: Training on generated data makes models forget. *arXiv preprint arXiv:2305.17493*, 2023.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. Ai models collapse when trained on recursively generated data. *Nature*, 631(8022):755–759, 2024.

David Silver and Richard S Sutton. Welcome to the era of experience. *Google AI*, 1, 2025.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

David Silver, Satinder Singh, Doina Precup, and Richard S Sutton. Reward is enough. *Artificial intelligence*, 299:103535, 2021.

SimpleVLA-RL Team. Simplevla-rl: Online rl with simple reward enables training vla models with only one trajectory. https://github.com/PRIME-RL/SimpleVLA-RL, 2025. GitHub repository.

Raghav Singhal, Zachary Horvitz, Ryan Teehan, Mengye Ren, Zhou Yu, Kathleen McKeown, and Rajesh Ranganath. A general framework for inference-time scaling and steering of diffusion models. In *Forty-second International Conference on Machine Learning*, 2025.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. R1-searcher: Incentivizing the search capability in llms via reinforcement learning. *arXiv preprint arXiv:2503.05592*, 2025a.

Huatong Song, Jinhao Jiang, Wenqing Tian, Zhipeng Chen, Yuhuan Wu, Jiahao Zhao, Yingqian Min, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. R1-searcher++: Incentivizing the dynamic knowledge acquisition of llms via reinforcement learning. *arXiv preprint arXiv:2505.17005*, 2025b.

Yuxuan Song, Zheng Zhang, Cheng Luo, Pengyang Gao, Fan Xia, Hao Luo, Zheng Li, Yuehang Yang, Hongli Yu, Xingwei Qu, et al. Seed diffusion: A large-scale diffusion language model with high-speed inference. *arXiv preprint arXiv:2508.02193*, 2025c.

Saksham Sahai Srivastava and Vaneet Aggarwal. A technical survey of reinforcement learning techniques for large language models. *arXiv preprint arXiv:2507.04136*, 2025.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021, 2020.

Zafir Stojanovski, Oliver Stanley, Joe Sharratt, Richard Jones, Abdulhakeem Adefioye, Jean Kaddour, and Andreas Köpf. Reasoning gym: Reasoning environments for reinforcement learning with verifiable rewards. *arXiv preprint arXiv:2505.24760*, 2025.

Alex Su, Haozhe Wang, Weiming Ren, Fangzhen Lin, and Wenhu Chen. Pixel reasoner: Incentivizing pixel-space reasoning with curiosity-driven reinforcement learning. *arXiv preprint arXiv:2505.15966*, 2025a.

Jinyan Su, Jennifer Healey, Preslav Nakov, and Claire Cardie. Between underthinking and overthinking: An empirical study of reasoning length and correctness in llms. *arXiv preprint arXiv:2505.00127*, 2025b.

Yi Su, Dian Yu, Linfeng Song, Juntao Li, Haitao Mi, Zhaopeng Tu, Min Zhang, and Dong Yu. Crossing the reward bridge: Expanding rl with verifiable rewards across diverse domains. *arXiv preprint arXiv:2503.23829*, 2025c.

Zhaochen Su, Linjie Li, Mingyang Song, Yunzhuo Hao, Zhengyuan Yang, Jun Zhang, Guanjie Chen, Jiawei Gu, Juntao Li, Xiaoye Qu, et al. Openthinkimg: Learning to think with images via visual tool reinforcement learning. *arXiv preprint arXiv:2505.08617*, 2025d.

Zhaochen Su, Peng Xia, Hangyu Guo, Zhenhua Liu, Yan Ma, Xiaoye Qu, Jiaqi Liu, Yanshu Li, Kaide Zeng, Zhengyuan Yang, et al. Thinking with images for multimodal reasoning: Foundations, methods, and future frontiers. *arXiv preprint arXiv:2506.23918*, 2025e.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Hanjie Chen, et al. Stop overthinking: A survey on efficient reasoning for large language models. *arXiv preprint arXiv:2503.16419*, 2025.

Hao Sun. Supervised fine-tuning as inverse reinforcement learning. *arXiv preprint arXiv:2403.12017*, 2024.

Hao Sun, Zile Qiao, Jiayan Guo, Xuanbo Fan, Yingyan Hou, Yong Jiang, Pengjun Xie, Yan Zhang, Fei Huang, and Jingren Zhou. Zerosearch: Incentivize the search capability of llms without searching. *arXiv preprint arXiv:2505.04588*, 2025a.

Jiankai Sun, Chuanyang Zheng, Enze Xie, Zhengying Liu, Ruihang Chu, Jianing Qiu, Jiaqi Xu, Mingyu Ding, Hongyang Li, Mengzhe Geng, et al. A survey of reasoning with foundation models: Concepts, methodologies, and outlook. *ACM Computing Surveys*, 57(11):1–43, 2025b.

Lin Sun, Chuang Liu, Xiaofeng Ma, Tao Yang, Weijia Lu, and Ning Wu. Freeprm: Training process reward models without ground truth process labels. *arXiv preprint arXiv:2506.03570*, 2025c.

Shengjie Sun, Runze Liu, Jiafei Lyu, Jing-Wen Yang, Liangpeng Zhang, and Xiu Li. A large language model-driven reward design framework via dynamic feedback for reinforcement learning. *Knowledge-Based Systems*, 326:114065, 2025d. ISSN 0950-7051. doi: https://doi.org/10.1016/j.knosys.2025.114065. URL https://www.sciencedirect.com/science/article/pii/S0950705125011104.

Yifan Sun, Jingyan Shen, Yibin Wang, Tianyu Chen, Zhendong Wang, Mingyuan Zhou, and Huan Zhang. Improving data efficiency for llm reinforcement fine-tuning through difficulty-targeted online data selection and rollout replay. *arXiv preprint arXiv:2506.05316*, 2025e.

Yu Sun, Xingyu Qian, Weiwen Xu, Hao Zhang, Chenghao Xiao, Long Li, Yu Rong, Wenbing Huang, Qifeng Bai, and Tingyang Xu. Reasonmed: A 370k multi-agent generated dataset for advancing medical reasoning. *arXiv preprint arXiv:2506.09513*, 2025f.

Zetian Sun, Dongfang Li, Zhuoen Chen, Yuhuai Qin, and Baotian Hu. Stabilizing long-term multi-turn reinforcement learning with gated rewards. *arXiv preprint arXiv:2508.10548*, 2025g.

Zhongxiang Sun, Qipeng Wang, Haoyu Wang, Xiao Zhang, and Jun Xu. Detection and mitigation of hallucination in large reasoning models: A mechanistic perspective. *arXiv preprint arXiv:2505.12886*, 2025h.

Zhoujian Sun, Ziyi Liu, Cheng Luo, Jiebin Chu, and Zhengxing Huang. Improving interactive diagnostic ability of a large language model agent through clinical experience learning, 2025i. URL https://arxiv.org/abs/2503.16463.

Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. Value-decomposition networks for cooperative multi-agent learning. *arXiv preprint arXiv:1706.05296*, 2017.

Richard S Sutton, Andrew G Barto, et al. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge, 1998.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

Gokul Swamy, Sanjiban Choudhury, Wen Sun, Zhiwei Steven Wu, and J Andrew Bagnell. All roads lead to likelihood: The value of reinforcement learning in fine-tuning. *arXiv preprint arXiv:2503.01067*, 2025.

Jaesung Tae, Hamish Ivison, Sachin Kumar, and Arman Cohan. Tess 2: A large-scale generalist diffusion language model. *arXiv preprint arXiv:2502.13917*, 2025.

Hongze Tan and Jianfei Pan. Gtpo and grpo-s: Token and sequence-level reward shaping with policy entropy. *arXiv preprint arXiv:2508.04349*, 2025.

Shuhan Tan, Kairan Dou, Yue Zhao, and Philipp Krähenbühl. Interactive post-training for vision-language-action models. *arXiv preprint arXiv:2505.17016*, 2025a.

Wentao Tan, Qiong Cao, Chao Xue, Yibing Zhan, Changxing Ding, and Xiaodong He. Chartmaster: Advancing chart-to-code generation with real-world charts and chart similarity reinforcement learning. *arXiv preprint arXiv:2508.17608*, 2025b.

Xiangru Tang, Tianrui Qin, Tianhao Peng, Ziyang Zhou, Daniel Shao, Tingting Du, Xinming Wei, Peng Xia, Fang Wu, He Zhu, et al. Agent kb: Leveraging cross-domain experience for agentic problem solving. *arXiv preprint arXiv:2507.06229*, 2025.

Zhengwei Tao, Jialong Wu, Wenbiao Yin, Junkai Zhang, Baixuan Li, Haiyang Shen, Kuan Li, Liwen Zhang, Xinyu Wang, Yong Jiang, et al. Webshaper: Agentically data synthesizing via information-seeking formalization. *arXiv preprint arXiv:2507.15061*, 2025.

ByteDance Seed Team. Seed-oss open-source models, 2025a. URL https://github.com/ByteDance-Seed/seed-oss.

Dolphin Team. Dolphin r1 dataset. https://huggingface.co/datasets/QuixiAI/dolphin-r1, 2025b. URL https://huggingface.co/datasets/QuixiAI/dolphin-r1. Dataset, Apache-2.0 license.

GLM-V. Team, Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, et al. GLM-4.5v and GLM-4.1v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning, 2025a.

Kimi Team. Kimi k2: Open agentic intelligence, 2025c. URL https://arxiv.org/abs/2507.20534.

Kimi Team. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025d.

MiroMind AI Team. Mirothinker: An open-source agentic model series trained for deep research and complex, long-horizon problem solving. https://github.com/MiroMindAI/MiroThinker, 2025e.

MiroMind Data Team. Miroverse v0.1: A reproducible, full-trajectory, ever-growing deep research dataset, 2025f. URL https://huggingface.co/datasets/miromind-ai/MiroVerse-v0.1.

Prime Intellect Team, Sami Jaghouar, Justus Mattern, Jack Min Ong, Jannik Straube, Manveer Basra, Aaron Pazdera, Kushal Thaman, Matthew Di Ferrante, Felix Gabriel, et al. Intellect-2: A reasoning model trained through globally decentralized reinforcement learning. *arXiv preprint arXiv:2505.07291*, 2025b.

Qwen Team. Qwq-32b: Embracing the power of reinforcement learning, March 2025g. URL https://qwenlm.github.io/blog/qwq-32b/.

RLinf Team. Rlinf: Reinforcement learning infrastructure for agentic ai. https://github.com/RLinf/RLinf, 2025h. GitHub repository.

Tencent Hunyuan Team, Ao Liu, Botong Zhou, Can Xu, Chayse Zhou, ChenChen Zhang, Chengcheng Xu, Chenhao Wang, Decheng Wu, Dengpeng Wu, et al. Hunyuan-turbos: Advancing large language models through mamba-transformer synergy and adaptive chain-of-thought. *arXiv preprint arXiv:2505.15431*, 2025c.

THUDM. slime: An sglang-native post-training framework for rl scaling. https://github.com/THUDM/slime, 2025. GitHub repository.

Shulin Tian, Ruiqi Wang, Hongming Guo, Penghao Wu, Yuhao Dong, Xiuying Wang, Jingkang Yang, Hao Zhang, Hongyuan Zhu, and Ziwei Liu. Ego-r1: Chain-of-tool-thought for ultra-long egocentric video reasoning. *arXiv preprint arXiv:2506.13654*, 2025.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

tokenbender. avatarl: training language models from scratch with pure reinforcement learning, 2025. URL https://github.com/tokenbender/avatarl.

Chengzhuo Tong, Ziyu Guo, Renrui Zhang, Wenyu Shan, Xinyu Wei, Zhenghao Xing, Hongsheng Li, and Pheng-Ann Heng. Delving into rl for image generation with cot: A study on dpo vs. grpo. *arXiv preprint arXiv:2505.17017*, 2025a.

Jingqi Tong, Jixin Tang, Hangcheng Li, Yurong Mou, Ming Zhang, Jun Zhao, Yanbo Wen, Fan Song, Jiahao Zhan, Yuyang Lu, et al. Code2logic: Game-code-driven data synthesis for enhancing vlms general reasoning. *arXiv preprint arXiv:2505.13886*, 2025b.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, et al. Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*, 2024.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. Appworld: A controllable world of apps and people for benchmarking interactive coding agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16022–16076, 2024.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022.

Carel van Niekerk, Renato Vukovic, Benjamin Matthias Ruppik, Hsien-chin Lin, and Milica Gašić. Post-training large language models via reinforcement learning from self-feedback. *arXiv preprint arXiv:2507.21931*, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Dheeraj Vattikonda, Santhoshi Ravichandran, Emiliano Penaloza, Hadi Nekoei, Megh Thakkar, Thibault Le Sellier de Chezelles, Nicolas Gontier, Miguel Muñoz-Mármol, Sahar Omidi Shayegan, Stefania Raimondo, et al. How to train your llm web agent: A statistical diagnosis. *arXiv preprint arXiv:2507.04103*, 2025.

Pablo Villalobos, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbhahn. Will we run out of data? limits of llm scaling based on human-generated data. *arXiv preprint arXiv:2211.04325*, 2022.

Vijay Viswanathan, Yanchao Sun, Shuang Ma, Xiang Kong, Meng Cao, Graham Neubig, and Tong-shuang Wu. Checklists are better than reward models for aligning language models. *arXiv preprint arXiv:2507.18624*, 2025.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. Trl: Transformer reinforcement learning. https://github.com/huggingface/trl, 2020.

Christian Walder and Deep Karkhanis. Pass@k policy optimization: Solving harder reinforcement learning problems. *arXiv preprint arXiv:2505.15201*, 2025.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8228–8238, 2024.

Ziyu Wan, Yunxiang Li, Xiaoyu Wen, Yan Song, Hanjing Wang, Linyi Yang, Mark Schmidt, Jun Wang, Weinan Zhang, Shuyue Hu, et al. Rema: Learning to meta-think for llms with multi-agent reinforcement learning. *arXiv preprint arXiv:2503.09501*, 2025.

Bin Wang, Bojun Wang, Changyi Wan, Guanzhe Huang, Hanpeng Hu, Haonan Jia, Hao Nie, Mingliang Li, Nuo Chen, Siyu Chen, et al. Step-3 is large yet affordable: Model-system co-design for cost-effective decoding. *arXiv preprint arXiv:2507.19427*, 2025a.

Chen Wang, Lai Wei, Yanzhi Zhang, Chenyang Shao, Zedong Dan, Weiran Huang, Yue Wang, and Yuzhi Zhang. Eframe: Deeper reasoning via exploration-filtering-replay reinforcement learning framework. *arXiv preprint arXiv:2506.22200*, 2025b.

Chenglong Wang, Yang Gan, Yifu Huo, Yongyu Mu, Qiaozhi He, Murun Yang, Bei Li, Tong Xiao, Chunliang Zhang, Tongran Liu, et al. Gram: A generative foundation reward model for reward generalization. *arXiv preprint arXiv:2506.14175*, 2025c.

Haiming Wang, Mert Unsal, Xiaohan Lin, Mantas Baksys, Junqi Liu, MD Santos, Flood Sung, Marina Vinyes, Zhenzhe Ying, Zekai Zhu, et al. Kimina-prover preview: Towards large formal reasoning models with reinforcement learning, 2025. *URL https://arxiv. org/abs/2504.11354*, 2025d.

Hanlin Wang, Chak Tou Leong, Jiashuo Wang, Jian Wang, and Wenjie Li. Spa-rl: Reinforcing llm agents via stepwise progress attribution. *arXiv preprint arXiv:2505.20732*, 2025e.

Hanyin Wang. Reinforcement learning for out-of-distribution reasoning in llms: An empirical study on diagnosis-related group coding, 2025. URL https://arxiv.org/abs/2505.21908.

Haoming Wang, Haoyang Zou, Huatong Song, Jiazhan Feng, Junjie Fang, Junting Lu, Longxiang Liu, Qinyu Luo, Shihao Liang, Shijue Huang, et al. Ui-tars-2 technical report: Advancing gui agent with multi-turn reinforcement learning. *arXiv preprint arXiv:2509.02544*, 2025f.

Haozhe Wang, Qixin Xu, Che Liu, Junhong Wu, Fangzhen Lin, and Wenhu Chen. Emergent hierarchical reasoning in llms through reinforcement learning. *arXiv preprint arXiv:2509.03646*, 2025g.

Jiakang Wang, Runze Liu, Fuzheng Zhang, Xiu Li, and Guorui Zhou. Stabilizing knowledge, promoting reasoning: Dual-token constraints for rlvr. *arXiv preprint arXiv:2507.15778*, 2025h.

Jicheng Wang, Yifeng He, and Hao Chen. Repogenreflex: Enhancing repository-level code completion with verbal reinforcement and retrieval-augmented generation. *arXiv preprint arXiv:2409.13122*, 2024a.

Junke Wang, Zhi Tian, Xun Wang, Xinyu Zhang, Weilin Huang, Zuxuan Wu, and Yu-Gang Jiang. Simplear: Pushing the frontier of autoregressive visual generation through pretraining, sft, and rl. *arXiv preprint arXiv:2504.11455*, 2025i.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce LLMs step-by-step without human annotations. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9426–9439, Bangkok, Thailand, August 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.510. URL https://aclanthology.org/2024.acl-long.510/.

Peiyu Wang, Yichen Wei, Yi Peng, Xiaokun Wang, Weijie Qiu, Wei Shen, Tianyidan Xie, Jiangbo Pei, Jianhao Zhang, Yunzhuo Hao, et al. Skywork r1v2: Multimodal hybrid reinforcement learning for reasoning, 2025j.

Qi Wang, Yanrui Yu, Ye Yuan, Rui Mao, and Tianfei Zhou. Videorft: Incentivizing video reasoning capability in mllms via reinforced fine-tuning. *arXiv preprint arXiv:2505.12434*, 2025k.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. Scienceworld: Is your agent smarter than a 5th grader? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11279–11298, 2022.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*, 2025l.

Weixun Wang, Shaopan Xiong, Gengru Chen, Wei Gao, Sheng Guo, Yancheng He, Ju Huang, Jiaheng Liu, Zhendong Li, Xiaoyang Li, et al. Reinforcement learning optimization for large-scale learning: An efficient and user-friendly scaling library. *arXiv preprint arXiv:2506.06122*, 2025m.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency, 2025n.

Xin Wang, Wenhan Xiong, Hongmin Wang, and William Yang Wang. Look before you leap: Bridging model-free and model-based reinforcement learning for planned-ahead vision-and-language navigation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 37–53, 2018.

Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6629–6638, 2019.

Yanlin Wang, Yanli Wang, Daya Guo, Jiachi Chen, Ruikai Zhang, Yuchi Ma, and Zibin Zheng. Rlcoder: Reinforcement learning for repository-level code completion. *arXiv preprint arXiv:2407.19487*, 2024c.

Yinjie Wang, Ling Yang, Bowen Li, Ye Tian, Ke Shen, and Mengdi Wang. Revolutionizing reinforcement learning framework for diffusion large language models. *arXiv preprint arXiv:2509.06949*, 2025o.

Yinjie Wang, Ling Yang, Ye Tian, Ke Shen, and Mengdi Wang. Co-evolving llm coder and unit tester via reinforcement learning. *arXiv preprint arXiv:2506.03136*, 2025p.

Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Liyuan Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, et al. Reinforcement learning for reasoning in large language models with one training example. *arXiv preprint arXiv:2504.20571*, 2025q.

Yiting Wang, Guoheng Sun, Wanghao Ye, Gang Qu, and Ang Li. Verireason: Reinforcement learning with testbench feedback for reasoning-enhanced verilog generation. *arXiv preprint arXiv:2505.11849*, 2025r.

Yue Wang, Qiuzhi Liu, Jiahao Xu, Tian Liang, Xingyu Chen, Zhiwei He, Linfeng Song, Dian Yu, Juntao Li, Zhuosheng Zhang, et al. Thoughts are all over the place: On the underthinking of o1-like llms. *arXiv preprint arXiv:2501.18585*, 2025s.

Zengzhi Wang, Fan Zhou, Xuefeng Li, and Pengfei Liu. Octothinker: Mid-training incentivizes reinforcement learning scaling. *arXiv preprint arXiv:2506.20512*, 2025t.

Zhilin Wang, Zhe Yang, Yun Luo, Yafu Li, Haoran Zhang, Runzhe Zhan, Derek F Wong, Jizhe Zhou, and Yu Cheng. Synthesizing sheet music problems for evaluation and reinforcement learning. *arXiv preprint arXiv:2509.04059*, 2025u.

Ziliang Wang, Xuhui Zheng, Kang An, Cijun Ouyang, Jialu Cai, Yuhang Wang, and Yichao Wu. Stepsearch: Igniting llms search ability via step-wise proximal policy optimization. *arXiv preprint arXiv:2505.15107*, 2025v.

Yuyang Wanyan, Xi Zhang, Haiyang Xu, Haowei Liu, Junyang Wang, Jiabo Ye, Yutong Kou, Ming Yan, Fei Huang, Xiaoshan Yang, et al. Look before you leap: A gui-critic-r1 model for pre-operative error diagnosis in gui automation. *arXiv preprint arXiv:2506.04614*, 2025.

Jason Wei. The asymmetry of verification and verifier's law. https://www.jasonwei.net/blog/asymmetry-of-verification-and-verifiers-law, 2025. Accessed: 2025-07-15.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. Browsecomp: A simple yet challenging benchmark for browsing agents. *arXiv preprint arXiv:2504.12516*, 2025a.

Yifan Wei, Xiaoyan Yu, Yixuan Weng, Tengfei Pan, Angsheng Li, and Li Du. Autotir: Autonomous tools integrated reasoning via reinforcement learning. *arXiv preprint arXiv:2507.21836*, 2025b.

Yuxiang Wei, Olivier Duchenne, Jade Copet, Quentin Carbonneaux, Lingming Zhang, Daniel Fried, Gabriel Synnaeve, Rishabh Singh, and Sida I Wang. Swe-rl: Advancing llm reasoning via reinforcement learning on open software evolution. *arXiv preprint arXiv:2502.18449*, 2025c.

Zhepei Wei, Wenlin Yao, Yao Liu, Weizhi Zhang, Qin Lu, Liang Qiu, Changlong Yu, Puyang Xu, Chao Zhang, Bing Yin, et al. Webagent-r1: Training web agents via end-to-end multi-turn reinforcement learning. *arXiv preprint arXiv:2505.16421*, 2025d.

Hao Wen, Xinrui Wu, Yi Sun, Feifei Zhang, Liye Chen, Jie Wang, Yunxin Liu, Ya-Qin Zhang, and Yuanchun Li. Budgetthinker: Empowering budget-aware llm reasoning with control tokens. *arXiv preprint arXiv:2508.17196*, 2025a.

Liang Wen, Yunke Cai, Fenrui Xiao, Xin He, Qi An, Zhenyu Duan, Yimin Du, Junchen Liu, Lifu Tang, Xiaowei Lv, et al. Light-r1: Curriculum sft, dpo and rl for long cot from scratch and beyond. *arXiv preprint arXiv:2503.10460*, 2025b.

Xumeng Wen, Zihan Liu, Shun Zheng, Zhijian Xu, Shengyu Ye, Zhirong Wu, Xiao Liang, Yang Wang, Junjie Li, Ziming Miao, et al. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base llms. *arXiv preprint arXiv:2506.14245*, 2025c.

Chenxi Whitehouse, Tianlu Wang, Ping Yu, Xian Li, Jason Weston, Ilia Kulikov, and Swarnadeep Saha. J1: Incentivizing thinking in llm-as-a-judge via reinforcement learning. *arXiv preprint arXiv:2505.10320*, 2025.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

Ronald J Williams and Jing Peng. Function optimization using connectionist reinforcement learning algorithms. *Connection Science*, 3(3):241–268, 1991.

Maciej Wołczyk, Michał Zając, Razvan Pascanu, Łukasz Kuciński, and Piotr Miłoś. Continual world: A robotic benchmark for continual reinforcement learning. *Advances in Neural Information Processing Systems*, 34:28496–28510, 2021.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025a.

Diankun Wu, Fangfu Liu, Yi-Hsin Hung, and Yueqi Duan. Spatial-mllm: Boosting mllm capabilities in visual-based spatial intelligence. *arXiv preprint arXiv:2505.23747*, 2025b.

Haoyuan Wu, Xueyi Chen, Rui Ming, Jilong Gao, Shoubo Hu, Zhuolun He, and Bei Yu. Totrl: Unlock llm tree-of-thoughts reasoning potential through puzzles solving. *arXiv preprint arXiv:2505.12717*, 2025c.

Jialong Wu, Baixuan Li, Runnan Fang, Wenbiao Yin, Liwen Zhang, Zhengwei Tao, Dingchu Zhang, Zekun Xi, Gang Fu, Yong Jiang, et al. Webdancer: Towards autonomous information seeking agency. *arXiv preprint arXiv:2505.22648*, 2025d.

Lixin Wu, Na Cai, Qiao Cheng, Jiachen Wang, and Yitao Duan. Confucius3-math: A lightweight high-performance reasoning llm for chinese k-12 mathematics learning. *arXiv preprint arXiv:2506.18330*, 2025e.

Mingrui Wu, Lu Wang, Pu Zhao, Fangkai Yang, Jianjin Zhang, Jianfeng Liu, Yuefeng Zhan, Weihao Han, Hao Sun, Jiayi Ji, et al. Reprompt: Reasoning-augmented reprompting for text-to-image generation via reinforcement learning. *arXiv preprint arXiv:2505.17540*, 2025f.

Penghao Wu, Shengnan Ma, Bo Wang, Jiaheng Yu, Lewei Lu, and Ziwei Liu. Gui-reflection: Empowering multimodal gui models with self-reflection behavior. *arXiv preprint arXiv:2506.08012*, 2025g.

Tianhao Wu, Weizhe Yuan, Olga Golovneva, Jing Xu, Yuandong Tian, Jiantao Jiao, Jason Weston, and Sainbayar Sukhbaatar. Meta-rewarding language models: Self-improving alignment with llm-as-a-meta-judge. *arXiv preprint arXiv:2407.19594*, 2024.

Weijia Wu, Chen Gao, Joya Chen, Kevin Qinghong Lin, Qingwei Meng, Yiming Zhang, Yuke Qiu, Hong Zhou, and Mike Zheng Shou. Reinforcement learning in vision: A survey. *arXiv preprint arXiv:2508.08189*, 2025h.

Xiaobao Wu. Sailing by the stars: A survey on reward models and learning strategies for learning from rewards. *arXiv preprint arXiv:2505.02686*, 2025.

Yongliang Wu, Yizhou Zhou, Zhou Ziheng, Yingzhe Peng, Xinyu Ye, Xinting Hu, Wenbo Zhu, Lu Qi, Ming-Hsuan Yang, and Xu Yang. On the generalization of sft: A reinforcement learning perspective with reward rectification. *arXiv preprint arXiv:2508.05629*, 2025i.

Jiaer Xia, Yuhang Zang, Peng Gao, Yixuan Li, and Kaiyang Zhou. Visionary-r1: Mitigating shortcuts in visual reasoning with reinforcement learning. *arXiv preprint arXiv:2505.14677*, 2025a.

Peng Xia, Jinglu Wang, Yibo Peng, Kaide Zeng, Xian Wu, Xiangru Tang, Hongtu Zhu, Yun Li, Shujie Liu, Yan Lu, et al. Mmedagent-rl: Optimizing multi-agent collaboration for multimodal medical reasoning. *arXiv preprint arXiv:2506.00555*, 2025b.

Yu Xia, Rui Wang, Xu Liu, Mingyan Li, Tong Yu, Xiang Chen, Julian McAuley, and Shuai Li. Beyond chain-of-thought: A survey of chain-of-x paradigms for llms. *arXiv preprint arXiv:2404.15676*, 2024.

Yunhui Xia, Wei Shen, Yan Wang, Jason Klein Liu, Huifeng Sun, Siyue Wu, Jian Hu, and Xiaolong Xu. Leetcodedataset: A temporal dataset for robust evaluation and efficient training of code llms. *arXiv preprint arXiv:2504.14655*, 2025c.

Violet Xiang, Chase Blagden, Rafael Rafailov, Nathan Lile, Sang Truong, Chelsea Finn, and Nick Haber. Just enough thinking: Efficient reasoning with adaptive length penalties reinforcement learning. *arXiv preprint arXiv:2506.05256*, 2025.

Changyi Xiao, Mengdi Zhang, and Yixin Cao. Bnpo: Beta normalization policy optimization. *arXiv preprint arXiv:2506.02864*, 2025a.

Teng Xiao, Yige Yuan, Mingxiao Li, Zhengyu Chen, and Vasant G Honavar. On a connection between imitation learning and rlhf. *arXiv preprint arXiv:2503.05079*, 2025b.

LLM Xiaomi, Bingquan Xia, Bowen Shen, Dawei Zhu, Di Zhang, Gang Wang, Hailin Zhang, Huaqiu Liu, Jiebao Xiao, Jinhao Dong, et al. Mimo: Unlocking the reasoning potential of language model–from pretraining to posttraining. *arXiv preprint arXiv:2505.07608*, 2025.

Chengxing Xie, Bowen Li, Chang Gao, He Du, Wai Lam, Difan Zou, and Kai Chen. Swe-fixer: Training open-source llms for effective and efficient github issue resolution. *arXiv preprint arXiv:2501.05040*, 2025a.

Guofu Xie, Yunsheng Shi, Hongtao Tian, Ting Yao, and Xiao Zhang. Capo: Towards enhancing llm reasoning through verifiable generative credit assignment. *arXiv preprint arXiv:2508.02298*, 2025b.

Tian Xie, Zitian Gao, Qingnan Ren, Haoming Luo, Yuqian Hong, Bryan Dai, Joey Zhou, Kai Qiu, Zhirong Wu, and Chong Luo. Logic-rl: Unleashing llm reasoning with rule-based reinforcement learning. *arXiv preprint arXiv:2502.14768*, 2025c.

Tianbao Xie, Siheng Zhao, Chen Henry Wu, Yitao Liu, Qian Luo, Victor Zhong, Yanchao Yang, and Tao Yu. Text2reward: Reward shaping with language models for reinforcement learning. *arXiv preprint arXiv:2309.11489*, 2023.

Yunfei Xie, Yinsong Ma, Shiyi Lan, Alan Yuille, Junfei Xiao, and Chen Wei. Play to generalize: Learning to reason through game play. *arXiv preprint arXiv:2506.08011*, 2025d.

Zhihui Xie, Liyu Chen, Weichao Mao, Jingjing Xu, Lingpeng Kong, et al. Teaching language models to critique via reinforcement learning. *arXiv preprint arXiv:2502.03492*, 2025e.

Zhihui Xie, Jiacheng Ye, Lin Zheng, Jiahui Gao, Jingwei Dong, Zirui Wu, Xueliang Zhao, Shansan Gong, Xin Jiang, Zhenguo Li, et al. Dream-coder 7b: An open diffusion language model for code. *arXiv preprint arXiv:2509.01142*, 2025f.

Rihui Xin, Han Liu, Zecheng Wang, Yupeng Zhang, Dianbo Sui, Xiaolin Hu, and Bingning Wang. Surrogate signals from format and length: Reinforcement learning for solving mathematical problems without ground truth answers. *arXiv preprint arXiv:2505.19439*, 2025.

Wei Xiong, Jiarui Yao, Yuhui Xu, Bo Pang, Lei Wang, Doyen Sahoo, Junnan Li, Nan Jiang, Tong Zhang, Caiming Xiong, et al. A minimalist approach to llm reasoning: from rejection sampling to reinforce. *arXiv preprint arXiv:2504.11343*, 2025a.

Wei Xiong, Hanning Zhang, Chenlu Ye, Lichang Chen, Nan Jiang, and Tong Zhang. Self-rewarding correction for mathematical reasoning. *arXiv preprint arXiv:2502.19613*, 2025b.

Fengli Xu, Qianyue Hao, Zefang Zong, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, Jiahui Gong, Tianjian Ouyang, Fanjin Meng, et al. Towards large reasoning models: A survey of reinforced reasoning with large language models. *arXiv preprint arXiv:2501.09686*, 2025a.

Huihui Xu and Yuanpeng Nie. Medground-r1: Advancing medical image grounding via spatial-semantic rewarded group relative policy optimization, 2025. URL https://arxiv.org/abs/2507.02994.

Ran Xu, Yuchen Zhuang, Yishan Zhong, Yue Yu, Xiangru Tang, Hang Wu, May D Wang, Peifeng Ruan, Donghan Yang, Tao Wang, et al. Medagentgym: Training llm agents for code-based medical reasoning at scale. *arXiv preprint arXiv:2506.04405*, 2025b.

Wenyuan Xu, Xiaochen Zuo, Chao Xin, Yu Yue, Lin Yan, and Yonghui Wu. A unified pairwise framework for rlhf: Bridging generative reward modeling and policy optimization. *arXiv preprint arXiv:2504.04950*, 2025c.

Wujiang Xu, Kai Mei, Hang Gao, Juntao Tan, Zujie Liang, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. *arXiv preprint arXiv:2502.12110*, 2025d.

Yi Xu, Chengzu Li, Han Zhou, Xingchen Wan, Caiqi Zhang, Anna Korhonen, and Ivan Vulić. Visual planning: Let's think only with images. *arXiv preprint arXiv:2505.11409*, 2025e.

Yuyang Xu, Yi Cheng, Haochao Ying, Zhuoyun Du, Renjun Hu, Xing Shi, Wei Lin, and Jian Wu. Sspo: Self-traced step-wise preference optimization for process supervision and reasoning compression. *arXiv preprint arXiv:2508.12604*, 2025f.

Zhangchen Xu, Yuetai Li, Fengqing Jiang, Bhaskar Ramasubramanian, Luyao Niu, Bill Yuchen Lin, and Radha Poovendran. Tinyv: Reducing false negatives in verification improves rl for llm reasoning. *arXiv preprint arXiv:2505.14625*, 2025g.

Zhangchen Xu, Yang Liu, Yueqin Yin, Mingyuan Zhou, and Radha Poovendran. Kodcode: A diverse, challenging, and verifiable synthetic dataset for coding. *arXiv preprint arXiv:2503.02951*, 2025h.

Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, et al. Dancegrpo: Unleashing grpo on visual generation. *arXiv preprint arXiv:2505.07818*, 2025.

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. Learning to reason under off-policy guidance. *arXiv preprint arXiv:2504.14945*, 2025a.

Kaiwen Yan, Xuanqing Shi, Hongcheng Guo, Wenxuan Wang, Zhuosheng Zhang, and Chengwei Qin. Drqa: Dynamic reasoning quota allocation for controlling overthinking in reasoning large language models. *arXiv preprint arXiv:2508.17803*, 2025b.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024a.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arxiv preprint arXiv: 2505.09388*, 2025a.

Chenyu Yang, Shiqian Su, Shi Liu, Xuan Dong, Yue Yu, Weijie Su, Xuehui Wang, Zhaoyang Liu, Jinguo Zhu, Hao Li, et al. Zerogui: Automating online gui learning at zero human cost. *arXiv preprint arXiv:2505.23762*, 2025b.

Jihan Yang, Shusheng Yang, Anjali W Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10632–10643, 2025c.

Kai Yang, Jian Tao, Jiafei Lyu, Chunjiang Ge, Jiaxin Chen, Weihan Shen, Xiaolong Zhu, and Xiu Li. Using human feedback to fine-tune diffusion models without any reward model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8941–8951, 2024b.

Ling Yang, Ye Tian, Bowen Li, Xinchen Zhang, Ke Shen, Yunhai Tong, and Mengdi Wang. Mmada: Multimodal large diffusion language models. *arXiv preprint arXiv:2505.15809*, 2025d.

Wenjie Yang, Mao Zheng, Mingyang Song, Zheng Li, and Sitong Wang. Ssr-zero: Simple self-rewarding reinforcement learning for machine translation. *arXiv preprint arXiv:2505.16637*, 2025e.

Yaodong Yang, Rui Luo, Minne Li, Ming Zhou, Weinan Zhang, and Jun Wang. Mean field multi-agent reinforcement learning. In *International conference on machine learning*, pages 5571–5580. PMLR, 2018.

Zhicheng Yang, Zhijiang Guo, Yinya Huang, Xiaodan Liang, Yiwei Wang, and Jing Tang. Treerpo: Tree relative policy optimization. *arXiv preprint arXiv:2506.05183*, 2025f.

Zhicheng Yang, Zhijiang Guo, Yinya Huang, Yongxin Wang, Dongchun Xie, Yiwei Wang, Xiaodan Liang, and Jing Tang. Depth-breadth synergy in rlvr: Unlocking llm reasoning gains with adaptive exploration. *arXiv preprint arXiv:2508.13755*, 2025g.

Zongxian Yang, Jiayu Qian, Zegao Peng, Haoyu Zhang, and Zhi-An Huang. Med-refl: Medical reasoning enhancement via self-corrected fine-grained reflection, 2025h. URL https://arxiv.org/abs/2506.13793.

Feng Yao, Liyuan Liu, Dinghuai Zhang, Chengyu Dong, Jingbo Shang, and Jianfeng Gao. Your efficient rl framework secretly brings you off-policy rl training, August 2025a. URL https://fengyao.notion.site/off-policy-rl.

Feng Yao, Zilong Wang, Liyuan Liu, Junxia Cui, Li Zhong, Xiaohan Fu, Haohui Mai, Vish Krishnan, Jianfeng Gao, and Jingbo Shang. Training language models to generate quality code with program analysis feedback. *arXiv preprint arXiv:2505.22704*, 2025b.

Chenlu Ye, Zhou Yu, Ziji Zhang, Hao Chen, Narayanan Sadagopan, Jing Huang, Tong Zhang, and Anurag Beniwal. Beyond correctness: Harmonizing process and outcome rewards through rl training. *arXiv preprint arXiv:2509.03403*, 2025a.

Jiabo Ye, Xi Zhang, Haiyang Xu, Haowei Liu, Junyang Wang, Zhaoqing Zhu, Ziwei Zheng, Feiyu Gao, Junjie Cao, Zhengxi Lu, et al. Mobile-agent-v3: Foundamental agents for gui automation. *arXiv preprint arXiv:2508.15144*, 2025b.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models. *arXiv preprint arXiv:2508.15487*, 2025c.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. Limo: Less is more for reasoning. *arXiv preprint arXiv:2502.03387*, 2025d.

Zhangyue Yin, Qiushi Sun, Zhiyuan Zeng, Qinyuan Cheng, Xipeng Qiu, and Xuanjing Huang. Dynamic and generalizable process reward modeling. *arXiv preprint arXiv:2507.17849*, 2025.

ByoungJun Jeon Yooseok Lim. More-clear: Multimodal offline reinforcement learning for clinical notes leveraged enhanced state representation, 2025. URL https://arxiv.org/abs/2508.07681.

Ailing Yu, Lan Yao, Jingnan Liu, Zhe Chen, Jiajun Yin, Yuan Wang, Xinhao Liao, Zhiling Ye, Ji Li, Yun Yue, et al. Medreseacher-r1: Expert-level medical deep researcher via a knowledge-informed trajectory synthesis framework. *arXiv preprint arXiv:2508.14880*, 2025a.

Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in neural information processing systems*, 35:24611–24624, 2022.

Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, et al. Memagent: Reshaping long-context llm with multi-conv rl-based memory agent. *arXiv preprint arXiv:2507.02259*, 2025b.

Hongzhou Yu, Tianhao Cheng, Yingwen Wang, Wen He, Qing Wang, Ying Cheng, Yuejie Zhang, Rui Feng, and Xiaobo Zhang. Finemedlm-o1: Enhancing medical knowledge reasoning ability of llm from supervised fine-tuning to test-time training, 2025c. URL https://arxiv.org/abs/2501.09213.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025d.

Tianyu Yu, Bo Ji, Shouli Wang, Shu Yao, Zefan Wang, Ganqu Cui, Lifan Yuan, Ning Ding, Yuan Yao, Zhiyuan Liu, et al. Rlpr: Extrapolating rlvr to general domains without verifiers. *arXiv preprint arXiv:2506.18254*, 2025e.

Zhaojian Yu, Yinghao Wu, Yilun Zhao, Arman Cohan, and Xiao-Ping Zhang. Z1: Efficient test-time scaling with code. *arXiv preprint arXiv:2504.00810*, 2025f.

Danlong Yuan, Tian Xie, Shaohan Huang, Zhuocheng Gong, Huishuai Zhang, Chong Luo, Furu Wei, and Dongyan Zhao. Efficient rl training for reasoning models via length-aware optimization. *arXiv preprint arXiv:2505.12284*, 2025a.

Jingyang Yuan, Huazuo Gao, Damai Dai, Junyu Luo, Liang Zhao, Zhengyan Zhang, Zhenda Xie, YX Wei, Lean Wang, Zhiping Xiao, et al. Native sparse attention: Hardware-aligned and natively trainable sparse attention. *arXiv preprint arXiv:2502.11089*, 2025b.

Lifan Yuan, Weize Chen, Yuchen Zhang, Ganqu Cui, Hanbin Wang, Ziming You, Ning Ding, Zhiyuan Liu, Maosong Sun, and Hao Peng. From f(x) and g(x) to f(g(x)): LLMs learn new skills in RL by composing old ones. https://husky-morocco-f72.notion.site/From-f-x-and-g-x-to-f-g-x-LLMs-Learn-New-Skills-in-RL-by-Composing-Old-Ones-2499aba4486f802c8108e76a12af3020, 2025c. Notion blog post, available online.

Lifan Yuan, Wendi Li, Huayu Chen, Ganqu Cui, Ning Ding, Kaiyan Zhang, Bowen Zhou, Zhiyuan Liu, and Hao Peng. Free process rewards without process labels. In *Forty-second International Conference on Machine Learning*, 2025d. URL https://openreview.net/forum?id=8ThnPFhGm8.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*, 3, 2024.

Weizhe Yuan, Jane Yu, Song Jiang, Karthik Padthe, Yang Li, Ilia Kulikov, Kyunghyun Cho, Dong Wang, Yuandong Tian, Jason E Weston, et al. Naturalreasoning: Reasoning in the wild with 2.8 m challenging questions. *arXiv preprint arXiv:2502.13124*, 2025e.

Yufeng Yuan, Yu Yue, Ruofei Zhu, Tiantian Fan, and Lin Yan. What's behind ppo's collapse in long-cot? value optimization holds the secret. *arXiv preprint arXiv:2503.01491*, 2025f.

Chuhuai Yue, Chengqi Dong, Yinan Gao, Hang He, Jiajun Chai, Guojun Yin, and Wei Lin. Promoting efficient reasoning with verifiable stepwise reward. *arXiv preprint arXiv:2508.10293*, 2025a.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025b.

Yu Yue, Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, TianTian Fan, Zhengyin Du, et al. Vapo: Efficient and reliable reinforcement learning for advanced reasoning tasks. *arXiv preprint arXiv:2504.05118*, 2025c.

Aohan Zeng, Xin Lv, Qinkai Zheng, Zhenyu Hou, Bin Chen, Chengxing Xie, Cunxiang Wang, Da Yin, Hao Zeng, Jiajie Zhang, et al. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models. *arXiv preprint arXiv:2508.06471*, 2025a.

Guangtao Zeng, Maohao Shen, Delin Chen, Zhenting Qi, Subhro Das, Dan Gutfreund, David Cox, Gregory Wornell, Wei Lu, Zhang-Wei Hong, et al. Satori-swe: Evolutionary test-time scaling for sample-efficient software engineering. *arXiv preprint arXiv:2505.23604*, 2025b.

Sihang Zeng, Kai Tian, Kaiyan Zhang, Junqi Gao, Runze Liu, Sa Yang, Jingxuan Li, Xinwei Long, Jiaheng Ma, Biqing Qi, et al. Reviewrl: Towards automated scientific review with rl. *arXiv preprint arXiv:2508.10308*, 2025c.

Siliang Zeng, Quan Wei, William Brown, Oana Frunza, Yuriy Nevmyvaka, and Mingyi Hong. Reinforcing multi-turn reasoning in llm agents via turn-level credit assignment. *arXiv preprint arXiv:2505.11821*, 2025d.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *arXiv preprint arXiv:2503.18892*, 2025e.

Kaiwen Zha, Zhengqi Gao, Maohao Shen, Zhang-Wei Hong, Duane S Boning, and Dina Katabi. Rl tango: Reinforcing generator and verifier together for language reasoning. *arXiv preprint arXiv:2505.15034*, 2025.

Bin Zhang, Hangyu Mao, Jingqing Ruan, Ying Wen, Yang Li, Shao Zhang, Zhiwei Xu, Dapeng Li, Ziyue Li, Rui Zhao, et al. Controlling large language model-based agents for large-scale decision-making: An actor-critic approach. *arXiv preprint arXiv:2311.13884*, 2023a.

Ceyao Zhang, Kaijie Yang, Siyi Hu, Zihao Wang, Guanghe Li, Yihang Sun, Cheng Zhang, Zhaowei Zhang, Anji Liu, Song-Chun Zhu, et al. Proagent: Building proactive cooperative ai with large language models. *CoRR*, 2023b.

Chong Zhang, Yue Deng, Xiang Lin, Bin Wang, Dianwen Ng, Hai Ye, Xingxuan Li, Yao Xiao, Zhanfeng Mo, Qi Zhang, et al. 100 days after deepseek-r1: A survey on replication studies and more directions for reasoning language models. *arXiv preprint arXiv:2505.00551*, 2025a.

Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin, Zaibin Zhang, Zelin Tan, Heng Zhou, Zhongzhi Li, Xiangyuan Xue, Yijiang Li, Yifan Zhou, Yang Chen, Chen Zhang, Yutao Fan, Zihu Wang, Songtao Huang, Yue Liao, Hongru Wang, Mengyue Yang, Heng Ji, Michael Littman, Jun Wang, Shuicheng Yan, Philip Torr, and Lei Bai. The landscape of agentic reinforcement learning for llms: A survey, 2025b. URL https://arxiv.org/abs/2509.02547.

Hongzhi Zhang, Jia Fu, Jingyuan Zhang, Kai Fu, Qi Wang, Fuzheng Zhang, and Guorui Zhou. Rlep: Reinforcement learning with experience replay for llm reasoning. *arXiv preprint arXiv:2507.07451*, 2025c.

Kaiyan Zhang, Runze Liu, Xuekai Zhu, Kai Tian, Sihang Zeng, Guoli Jia, Yuchen Fan, Xingtai Lv, Yuxin Zuo, Che Jiang, Ziyang Liu, Jianyu Wang, Yuru Wang, Ruotong Zhao, Ermo Hua, Yibo Wang, Shijie Wang, Junqi Gao, Xinwei Long, Youbang Sun, Zhiyuan Ma, Ganqu Cui, Lei Bai, Ning Ding, Biqing Qi, and Bowen Zhou. Marti: A framework for multi-agent llm systems reinforced training and inference, 2025d. URL https://github.com/TsinghuaC3I/MARTI.

Kaiyan Zhang, Jiayuan Zhang, Haoxin Li, Xuekai Zhu, Ermo Hua, Xingtai Lv, Ning Ding, Biqing Qi, and Bowen Zhou. OpenPRM: Building open-domain process-based reward models with preference trees. In *The Thirteenth International Conference on Learning Representations*, 2025e. URL https://openreview.net/forum?id=fGIqGfmgkW.

Kaiyi Zhang, Ang Lv, Jinpeng Li, Yongbo Wang, Feng Wang, Haoyuan Hu, and Rui Yan. Stephint: Multi-level stepwise hints enhance reinforcement learning to reason. *arXiv preprint arXiv:2507.02841*, 2025f.

Kongcheng Zhang, Qi Yao, Shunyu Liu, Yingjie Wang, Baisheng Lai, Jieping Ye, Mingli Song, and Dacheng Tao. Consistent paths lead to truth: Self-rewarding reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.08745*, 2025g.

Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. Generative verifiers: Reward modeling as next-token prediction. *arXiv preprint arXiv:2408.15240*, 2024a.

Qingyang Zhang, Haitao Wu, Changqing Zhang, Peilin Zhao, and Yatao Bian. Right question is already half the answer: Fully unsupervised llm reasoning incentivization. *arXiv preprint arXiv:2504.05812*, 2025h.

Ruize Zhang, Zelai Xu, Chengdong Ma, Chao Yu, Wei-Wei Tu, Wenhao Tang, Shiyu Huang, Deheng Ye, Wenbo Ding, Yaodong Yang, et al. A survey on self-play methods in reinforcement learning. *arXiv preprint arXiv:2408.01072*, 2024b.

Sheng Zhang, Qianchu Liu, Guanghui Qin, Tristan Naumann, and Hoifung Poon. Med-rlvr: Emerging medical reasoning from a 3b base model via reinforcement learning. *arXiv preprint arXiv:2502.19655*, 2025i.

Wenhao Zhang, Yuexiang Xie, Yuchang Sun, Yanxi Chen, Guoyin Wang, Yaliang Li, Bolin Ding, and Jingren Zhou. On-policy rl meets off-policy experts: Harmonizing supervised fine-tuning and reinforcement learning via dynamic weighting. *arXiv preprint arXiv:2508.11408*, 2025j.

Xiaotian Zhang, Yuan Wang, Zhaopeng Feng, Ruizhe Chen, Zhijie Zhou, Yan Zhang, Hongxia Xu, Jian Wu, and Zuozhu Liu. Med-u1: Incentivizing unified medical reasoning in llms via large-scale reinforcement learning. *arXiv preprint arXiv:2506.12307*, 2025k.

Xiaoying Zhang, Hao Sun, Yipeng Zhang, Kaituo Feng, Chaochao Lu, Chao Yang, and Helen Meng. Critique-grpo: Advancing llm reasoning with natural language and numerical feedback. *arXiv preprint arXiv:2506.03106*, 2025l.

Xintong Zhang, Zhi Gao, Bofei Zhang, Pengxiang Li, Xiaowen Zhang, Yang Liu, Tao Yuan, Yuwei Wu, Yunde Jia, Song-Chun Zhu, et al. Chain-of-focus: Adaptive visual search and zooming for multimodal reasoning via rl. *arXiv preprint arXiv:2505.15436*, 2025m.

Xuanyu Zhang, Weiqi Li, Shijie Zhao, Junlin Li, Li Zhang, and Jian Zhang. Vq-insight: Teaching vlms for ai-generated video quality understanding via progressive visual reinforcement learning. *arXiv preprint arXiv:2506.18564*, 2025n.

Xuechen Zhang, Zijian Huang, Yingcong Li, Chenshun Ni, Jiasi Chen, and Samet Oymak. Bread: Branched rollouts from expert anchors bridge sft and rl for reasoning. *arXiv preprint arXiv:2506.17211*, 2025o.

Yanzhi Zhang, Zhaoxi Zhang, Haoxiang Guan, Yilin Cheng, Yitong Duan, Chen Wang, Yue Wang, Shuxin Zheng, and Jiyan He. No free lunch: Rethinking internal feedback for llm reasoning. *arXiv preprint arXiv:2506.17219*, 2025p.

Yifan Zhang, Yifeng Liu, Huizhuo Yuan, Yang Yuan, Quanquan Gu, and Andrew C Yao. On the design of kl-regularized policy gradient algorithms for llm reasoning. *arXiv preprint arXiv:2505.17508*, 2025q.

Yimeng Zhang, Tian Wang, Jiri Gesi, Ziyi Wang, Yuxuan Lu, Jiacheng Lin, Sinong Zhan, Vianne Gao, Ruochen Jiao, Junze Liu, et al. Shop-r1: Rewarding llms to simulate human behavior in online shopping via reinforcement learning. *arXiv preprint arXiv:2507.17842*, 2025r.

Yu Zhang, Yunqi Li, Yifan Yang, Rui Wang, Yuqing Yang, Dai Qi, Jianmin Bao, Dongdong Chen, Chong Luo, and Lili Qiu. Reasongen-r1: Cot for autoregressive image generation models through sft and rl. *arXiv preprint arXiv:2505.24875*, 2025s.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. The lessons of developing process reward models in mathematical reasoning. *arXiv preprint arXiv:2501.07301*, 2025t.

Zhong Zhang, Yaxi Lu, Yikun Fu, Yupeng Huo, Shenzhi Yang, Yesai Wu, Han Si, Xin Cong, Haotian Chen, Yankai Lin, et al. Agentcpm-gui: Building mobile-use agents with reinforcement fine-tuning. *arXiv preprint arXiv:2506.01391*, 2025u.

Zizhuo Zhang, Jianing Zhu, Xinmu Ge, Zihua Zhao, Zhanke Zhou, Xuan Li, Xiao Feng, Jiangchao Yao, and Bo Han. Co-reward: Self-supervised reinforcement learning for large language model reasoning via contrastive agreement. *arXiv preprint arXiv:2508.00410*, 2025v.

Andrew Zhao, Yiran Wu, Yang Yue, Tong Wu, Quentin Xu, Matthieu Lin, Shenzhi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. Absolute zero: Reinforced self-play reasoning with zero data. *arXiv preprint arXiv:2505.03335*, 2025a.

Jian Zhao, Runze Liu, Kaiyan Zhang, Zhimu Zhou, Junqi Gao, Dong Li, Jiafei Lyu, Zhouyi Qian, Biqing Qi, Xiu Li, et al. Genprm: Scaling test-time compute of process reward models via generative reasoning. *arXiv preprint arXiv:2504.00891*, 2025b.

Siyan Zhao, Devaansh Gupta, Qinqing Zheng, and Aditya Grover. d1: Scaling reasoning in diffusion large language models via reinforcement learning, 2025c. URL https://arxiv.org/abs/2504.12216.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2), 2023a.

Weikang Zhao, Xili Wang, Chengdi Ma, Lingbin Kong, Zhaohua Yang, Mingxiang Tuo, Xiaowei Shi, Yitao Zhai, and Xunliang Cai. Mua-rl: Multi-turn user-interacting agent reinforcement learning for agentic tool use. *arXiv preprint arXiv:2508.18669*, 2025d.

Xuandong Zhao, Zhewei Kang, Aosong Feng, Sergey Levine, and Dawn Song. Learning to reason without external rewards. *arXiv preprint arXiv:2505.19590*, 2025e.

Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, et al. Pytorch fsdp: experiences on scaling fully sharded data parallel. *arXiv preprint arXiv:2304.11277*, 2023b.

Yuzhong Zhao, Yue Liu, Junpeng Liu, Jingye Chen, Xun Wu, Yaru Hao, Tengchao Lv, Shaohan Huang, Lei Cui, Qixiang Ye, et al. Geometric-mean policy optimization. *arXiv preprint arXiv:2507.20673*, 2025f.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025a.

Haizhong Zheng, Yang Zhou, Brian R Bartoldson, Bhavya Kailkhura, Fan Lai, Jiawei Zhao, and Beidi Chen. Act only when it pays: Efficient reinforcement learning for llm reasoning via selective rollouts. *arXiv preprint arXiv:2506.02177*, 2025b.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

Tianyu Zheng, Tianshun Xing, Qingshui Gu, Taoran Liang, Xingwei Qu, Xin Zhou, Yizhi Li, Zhoufutu Wen, Chenghua Lin, Wenhao Huang, et al. First return, entropy-eliciting explore. *arXiv preprint arXiv:2507.07017*, 2025c.

Xuejing Zheng, Chao Yu, and Minjie Zhang. Lifelong reinforcement learning with temporal logic formulas and reward machines. *Knowledge-Based Systems*, 257:109650, 2022.

Yaowei Zheng, Junting Lu, Shenzhi Wang, Zhangchi Feng, Dongdong Kuang, and Yuwen Xiong. Easyr1: An efficient, scalable, multi-modality rl training framework. https://github.com/hiyouga/EasyR1, 2025d.

Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. Deepresearcher: Scaling deep research via reinforcement learning in real-world environments. *arXiv preprint arXiv:2504.03160*, 2025e.

Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. Deepeyes: Incentivizing" thinking with images" via reinforcement learning. *arXiv preprint arXiv:2505.14362*, 2025f.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory. In *38th AAAI Conference on Artificial Intelligence, AAAI 2024, Feb 20-27 2024 Vancouver, Canada*, volume 38, pages 19724–19731. Association for the Advancement of Artificial Intelligence (AAAI), 2024.

Yifan Zhong, Fengshuo Bai, Shaofei Cai, Xuchuan Huang, Zhang Chen, Xiaowei Zhang, Yuanfei Wang, Shaoyang Guo, Tianrui Guan, Ka Nam Lui, et al. A survey on vision-language-action models: An action tokenization perspective. *arXiv preprint arXiv:2507.01925*, 2025.

Enshen Zhou, Jingkun An, Cheng Chi, Yi Han, Shanyu Rong, Chi Zhang, Pengwei Wang, Zhongyuan Wang, Tiejun Huang, Lu Sheng, et al. Roborefer: Towards spatial referring with reasoning in vision-language models for robotics. *arXiv preprint arXiv:2506.04308*, 2025a.

Junting Zhou, Wang Li, Yiyan Liao, Nengyuan Zhang, Tingjia Miaoand Zhihui Qi, Yuhan Wu, and Tong Yang. Academicbrowse: Benchmarking academic browse ability of llms. *arXiv preprint arXiv:2506.13784*, 2025b.

Meng Zhou, Bei Li, Jiahao Liu, Xiaowen Shi, Yang Bai, Rongxiang Weng, Jingang Wang, and Xunliang Cai. Libra: Assessing and improving reward model by learning to think. *arXiv preprint arXiv:2507.21645*, 2025c.

Ruochen Zhou, Minrui Xu, Shiqi Chen, Junteng Liu, Yunqi Li, Xinxin Lin, Zhengyu Chen, and Junxian He. Does learning mathematical problem-solving generalize to broader reasoning? *arXiv preprint arXiv:2507.04391*, 2025d.

Xiangxin Zhou, Zichen Liu, Anya Sims, Haonan Wang, Tianyu Pang, Chongxuan Li, Liang Wang, Min Lin, and Chao Du. Reinforcing general reasoning without verifiers. *arXiv preprint arXiv:2505.21493*, 2025e.

Yang Zhou, Sunzhu Li, Shunyu Liu, Wenkai Fang, Jiale Zhao, Jingwen Yang, Jianwei Lv, Kongcheng Zhang, Yihe Zhou, Hengtong Lu, et al. Breaking the exploration bottleneck: Rubric-scaffolded reinforcement learning for general llm reasoning. *arXiv preprint arXiv:2508.16949*, 2025f.

Yifei Zhou, Song Jiang, Yuandong Tian, Jason Weston, Sergey Levine, Sainbayar Sukhbaatar, and Xian Li. Sweet-rl: Training multi-turn llm agents on collaborative reasoning tasks. *arXiv preprint arXiv:2503.15478*, 2025g.

Yuqi Zhou, Sunhao Dai, Shuai Wang, Kaiwen Zhou, Qinglin Jia, and Jun Xu. Gui-g1: Understanding r1-zero-like training for visual grounding in gui agents. *arXiv preprint arXiv:2505.15810*, 2025h.

Zijian Zhou, Ao Qu, Zhaoxuan Wu, Sunghwan Kim, Alok Prakash, Daniela Rus, Jinhua Zhao, Bryan Kian Hsiang Low, and Paul Pu Liang. Mem1: Learning to synergize memory and reasoning for efficient long-horizon agents. *arXiv preprint arXiv:2506.15841*, 2025i.

Dingwei Zhu, Shihan Dou, Zhiheng Xi, Senjie Jin, Guoqiang Zhang, Jiazheng Zhang, Junjie Ye, Mingxu Chai, Enyu Zhou, Ming Zhang, et al. Vrpo: Rethinking value modeling for robust rl training under noisy supervision. *arXiv preprint arXiv:2508.03058*, 2025a.

Fengqi Zhu, Rongzhen Wang, Shen Nie, Xiaolu Zhang, Chunwei Wu, Jun Hu, Jun Zhou, Jianfei Chen, Yankai Lin, Ji-Rong Wen, et al. Llada 1.5: Variance-reduced preference optimization for large language diffusion models. *arXiv preprint arXiv:2505.19223*, 2025b.

Hui Zhu, Xv Wang, Zhenyu Wang, and Kai Xv. An emotion-sensitive dialogue policy for task-oriented dialogue system. *Scientific Reports*, 14(1):19759, 2024.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. InternVL3: Exploring advanced training and test-time recipes for open-source multimodal models, 2025c.

Qin Zhu, Fei Huang, Runyu Peng, Keming Lu, Bowen Yu, Qinyuan Cheng, Xipeng Qiu, Xuanjing Huang, and Junyang Lin. Autologi: Automated generation of logic puzzles for evaluating reasoning abilities of large language models. *arXiv preprint arXiv:2502.16906*, 2025d.

Wenhong Zhu, Ruobing Xie, Rui Wang, Xingwu Sun, Di Wang, and Pengfei Liu. Proximal supervised fine-tuning. *arXiv preprint arXiv:2508.17784*, 2025e.

Yaoyu Zhu, Di Huang, Hanqi Lyu, Xiaoyun Zhang, Chongxiao Li, Wenxuan Shi, Yutong Wu, Jianan Mu, Jinghua Wang, Yang Zhao, et al. Codev-r1: Reasoning-enhanced verilog generation. *arXiv preprint arXiv:2505.24183*, 2025f.

Yekun Zhu, Guang Chen, and Chengjun Mao. Think in blocks: Adaptive reasoning from direct response to deep reasoning. *arXiv preprint arXiv:2508.15507*, 2025g.

Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pages 2165–2183. PMLR, 2023.

Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.

Jiaru Zou, Ling Yang, Jingwen Gu, Jiahao Qiu, Ke Shen, Jingrui He, and Mengdi Wang. Reasonflux-prm: Trajectory-aware prms for long chain-of-thought reasoning in llms. *arXiv preprint arXiv:2506.18896*, 2025.

Yuxin Zuo, Shang Qu, Yifei Li, Zhangren Chen, Xuekai Zhu, Ermo Hua, Kaiyan Zhang, Ning Ding, and Bowen Zhou. Medxpertqa: Benchmarking expert-level medical reasoning and understanding. *arXiv preprint arXiv:2501.18362*, 2025a.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen Zhang, Xinwei Long, Ermo Hua, et al. Ttrl: Test-time reinforcement learning. *arXiv preprint arXiv:2504.16084*, 2025b.

Adam Zweiger, Jyothish Pari, Han Guo, Ekin Akyürek, Yoon Kim, and Pulkit Agrawal. Self-adapting language models. *arXiv preprint arXiv:2506.10943*, 2025.